Meshless Local Petrov-Galerkin method with Rankine source solution for two-dimensional two-phase flow modelling

Yan Zhou

Supervisor

Prof. Qingwei Ma

A thesis submitted for the degree of Doctor of Philosophy

School of Engineering and Mathematical Sciences

City University, London

December, 2015

Contents

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Prof. Qingwei Ma for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

Besides my advisor, I would like to thank every member in the research group, especially to Dr. Shiqiang Yan, for their warm help in every detailed aspects of this study and for sharing together and encouraging each other.

I gratefully acknowledges the financial support of my four years' study provided by the China Scholarship Counsel.

Last but not the least, I would like to thank my parents and my boyfriend Hao Yang for always being with me and supporting me spiritually throughout the study and my life in general.

Declaration

"I hereby certify that the work embodied in this Thesis is the result of original research and has not been submitted for a higher degree to any other University or Institute"

_____

Yan Zhou

List of Figures

List of tables

Notation

$\boldsymbol{r}, \boldsymbol{x}$	particle position vector
$\boldsymbol{r}^*$	intermediate position vector of fluid
$\boldsymbol{u}$	velocity vector of fluid
$\boldsymbol{u}^*$	intermediate velocity vector of fluid
$\boldsymbol{u}_l^{**}$	correction velocity
$\boldsymbol{U}$	velocity vector of Rigid boundary
$\dot{\boldsymbol{U}}$	acceleration of rigid boundary
υ	viscosity of fluid
p	pressure
τ_n	normal shear stress
σ	interface tension coefficient
κ	curvature
ρ, ρ_1, ρ_2	fluid density
$\boldsymbol{g}$	gravitational acceleration
l, k	phase indicator
$\boldsymbol{F}_d$	drag force of isolated particle
C_d	drag coefficient
$\boldsymbol{n}$	normal direction
dt	time step
dl	particle distance
t	time
φ	test function
Δl	initial particle distance
r	distance between particles
R_I	radius of integral domain
R_e	radius of support domain
M, N, m, n	total particle number in support domain
Ω_I	integral domain of particle I
$\partial\Omega_I$	surface of integral domain of particle I
ϕ	shape function
$\boldsymbol{\psi}$	base function in MLS method

w	weight function
θ	angle parameter
f	a specific function
$f_{,x}, f_{,y}$	function derivative in x or y direction
$\boldsymbol{f}_{\sigma}$	Surface tension force
n_0	initial inner particle number density
n^*, n'	intermediate and correction particle number density
n_i	particle number density of particle I
β	absolute density gradient
β_0	characteristic value of the absolute density gradient
α, γ	upper and lower bound of the absolute density gradient
α_V	volume fraction
$\boldsymbol{K}$	coefficient matrix of pressure equation
$\mathbf{F}$	coefficient arrow of pressure equation
h_0, h_1, h_2	height of fluid filling
x_0	length of fluid filling
L	tank length
D	tank depth
δ	density ratio
Φ	velocity potential
ζ	wave elevation
ω	frequency
T	natural period
a, a_0	amplitude
c	phase indicator function
Re	Reynolds number
Bo	bond number
∇	gradient operator
∇^2	Laplacian operator
$\frac{\partial}{\partial t}$	partial time derivative
$\frac{D}{Dt}$	substantial derivative

Abstract

A Lagrangian particle multiphase model based on the Meshless Local Petrov-Galerkin method with Rankine source solution (MLPG_R) is proposed to simulate 2D flows of two immiscible fluids. The model is applicable to fluids with a wide range of density ratio from 1.01 to 1000, capable of dealing with violent flow situations (e.g. breaking waves) and maintaining the sharp discontinuity of properties of the fluids at the interface.

In order to extend the MLPG_R method to model multiphase flows, an innovative phase coupling is proposed using an equation for pressure at the interface particle through two stages. In the first stage, the formulation is based on ensuring the continuity of the pressure and the ratio of the pressure derivative in the normal direction of the interface to the fluid density across the interface. Gravity current, natural sloshing of two layered liquids and air-water violent sloshing are successfully simulated and compared with either experimental data or analytical solution demonstrating a second order convergent rate. The second stage involves the extension of the method to account for interface tension and large viscosity. This is achieved by adding additional terms in the original pressure formulation considering jumps for both the pressure and the ratio of pressure derivatives in either the normal or tangential direction to fluid density at the interface. The method ensures both velocity continuity even with highly viscous fluids and interface stress balance in the presence of interface tension. Simulation of square-droplet deformation illustrates sharp pressure drop at the interface and relieved spurious currents that are known to be associated with predictions by other existing models. The capillary wave case also demonstrates the necessity of maintaining the jump in the ratio of pressure gradient to fluid density and the bubble rising case further validates the model as compared with the benchmark numerical results.

Apart from being the first application of the MLPG_R method to multiphase flows the proposed model also contains two highly effective and robust techniques whose applicability is not restricted to the MLPG_R method. One is based on use of the absolute density gradient for identifying the interface and isolated particles which is essential to ensure that interface conditions are applied at the correct locations in violent flows. The effectiveness of the technique has been examined by a number of particle configurations, including those with different levels of randomness of particle distribution. The other is about solving the discretised pressure equation, by splitting the one set equations into two sets corresponding to two phases and solving them separately but coupled by the interface particles. This technique efficiently gives reasonable solutions not only for the cases with low density ratio but also for the ones with very high density ratio.

Chapter 1 Introduction

1.1 Background of the multiphase flow

Multiphase flow is a flow system consisting of different fluid phases immiscible to each other, or having fluid and solid phases simultaneously. In flows containing fluids only, the phases may be of the same substance such as a liquid and its vapour, or of different substances such as two liquids or a liquid and a gas. The compressibility for most liquids is negligible maintaining a constant density. But it might become considerable for their vapours and gases when they are laden with high pressure, leading to varied density for a certain phase. In a fluid and solid system, gases or liquids constitute the fluid domain which coexist with the dispersed solid. These classifications are by no means unique. In some works classifications are defined in terms of length scales of the flow (Hu & Adams 2006; Wörner 2012), sorting into micro-, meso- and macroscopic, dominated to different extents by inertial forces as well as forces between various phases caused by viscosity difference and interface tension. Some texts identify the multiphase flow with general topologies namely disperse flows consisting of finite particles (Delnoij et al. 1997; Prosperetti & Tryggvason 2007), drops or bubbles distributed in a continuous phase and separate flows with two or more continuous streams separated by interfaces.

In nature, the multiphase flow is ubiquitous, such as rainy or showery winds, air and water pollutions, volcanic activities, mud slides, sediment transport, the 'white water' produced by breaking waves, debris flows, and countless other natural phenomenon. Numerous industrial processing systems also involve multiphase flows including cavitating pumps and turbines, electrophotographic processes, combustion engines, propulsion systems, oil and gas production and transport, etc. Multiphase flows also exist in biological and medical industry, from blood flow to laser surgery cavitation and so on. Accurate

predictions of the flow behaviour of those processes are essential for achieving the desired efficiency and effectiveness of the systems.

Even though the term of 'multiphase flow' is well understood, however, due to its diversity and complexity, the choice of the methods to be used largely depends on the types of flows and constrains that are imposed on the flow as well as motivation of the investigation. This work aims to study only separate multiphase flows with continuous phases separated by a single or multiple interfaces. When dispersed flow particles occur occasionally, they are regarded as dilute and isolated. Based on the length scale classification, the study is limited to systems that are well described by continuum theories, i.e. equations of continuity and momentum conservation, without considering either electronic forces or random particle motions that belong to the domain of statistical mechanics.

1.2 Multiphase flow models

1.2.1 Analytical models

To predict the detailed behaviour and reveal the phenomena of multiphase flows, there exist mainly three approaches, namely theoretical, experimental and numerical. Purely analytical method is the most efficient and economical but is inevitably limited by the complexity of the multiphase flow. As for solid bodies in the fluid, even single sphere solutions are limited to very small or very large Reynolds number. The helpfulness of the analytical method is reduced further when flows become general with intermediate Reynolds number or when more bodies interact. When two continuous fluids are present, solutions are usually restricted to small deformations even for the simplest case by ignoring either interface tensions or viscosities which may play dominant roles at certain times. For example the development of Rayleigh-Taylor instability is solved considering the interface tension but not viscosity in Drazin & Reid (1981) and later the correction took the viscosity into consideration in Velazquez et al. (1998). Both solutions are feasible at the very beginning of the instability but fail to predict the flow development when the configuration becomes complex. Another example is the natural sloshing of layered fluids (Faltinsen & Timokha 2009) in which, the linear solution of natural frequencies and corresponding modes for zero tank excitation are provided. The predictive capability of such solutions is limited due to ignoring the effects of viscosity and interface tension. When it comes to violent sloshing caused by the external excitation,

the analytical solution become either far from satisfactory due to various grossly unrealistic assumptions imposed on the flow or simply impossible to obtain.

1.2.2 Experimental models

Through laboratory scale models equipped with appropriate instrumentation, numerous experimental studies involving various types of multiphase flows, e.g. sedimentation processes (Komatina & Jovanovic 1997), single bubble rising (Bhaga & Weber 1981), bubble column (Walke & Sathe 2012), fluidization (Xue et al. 2012), flow in porous media (Das & Mirzaei 2012), lock exchange (Lowe et al. 2005) etc., have been carried out due to its general reliability and ease of data acquisition. However, apart from their high cost, there are also cases for which laboratory models are impossible to conduct or insufficient to acquire desired parameters. For example in the bubble column case, precise characterization of the bubble sizes and distributions is hard to achieve; also for the breaking wave case, even with high speed and high resolution cameras, the visualization of the entrained air is problematic, let alone accurate determination of continuous changing flow volume and size distribution, with 'white water' being taken only as snapshots. Perhaps the most important issue encountered by the experimental model is the differences in scales between experimental models and the prototype, which often would require a theoretical or computational model to extrapolate the laboratory results to that of the prototype scale reliably.

1.2.3 Numerical models

Numerical models have become an essential tool for the investigation of multiphase flow due to their ability to consider various effects arising in multiphase flows, such as interface tension, density and viscosity differences, complex interface configurations etc. Besides, numerical models are more promising to solve the issues of scale and visualization of any desired parameter. A diverse range of numerical models have been proposed for different types of the multiphase flow of which an important branch is the dispersed flow. To deal with complex flow configurations, most numerical models are based on Navier-Stokes equations, either Direct Numerical Simulation (DNS) or with turbulent models, solved by imposing an equation of state to identify the relation between pressure and density for compressible fluids or separating velocity and pressure to form the Poisson's equation for incompressible fluids. To simulate the dispersed phase, two types of models are prevalent, trajectory models and two-fluid models. The former

simulates the moving of either actual particles or larger, representative particles driven by drag, dynamic pressure gradient and inter-particle collision forces and the momentum equation of the continuous phase is augmented by point forces which represented the effect of the particles. Examples are droplets near the nozzle zone of a spray dryer (Gauvin et al. 1975), gas bubble rising in water column (Buwa et al. 2006; Delnoij et al. 1997) and sedimentation of solid particles in fluid (Patankar & Joseph 2001). Even though artificial collision models between particles are proposed and momentum transfer between discrete phase and continuous phase is considered, this model is usually restricted to dilute dispersed flows in which particle-particle interactions can be neglected. The alternative two-fluid model considers the interactions between the dispersed phase by treating it as a continuous phase and neglecting its discrete nature. This model is more applicable to predict dense particle/droplet flow where the interactions within the dispersed phase are dominant and has higher efficiency in dealing with large number of individual particles such as high bubble-laden flows (Druzhinin & Elghobashi 1998) and fluidized bed (Xue et al. 2012).

The flow type that is focused on in this thesis is another category of multiphase flows generally known as separate flows, which is mainly solved by single phase flow equations in two streams, coupled through appropriate dynamic and kinematic conditions at the interface. For solving the Navier-Stokes equations, fixed mesh models adopting either finite element method or finite volume method were first developed for the multiphase simulation (Bothe et al. 2011; Fedkiw et al. 1999; Hysing et al. 2009; Rudman 1997). These models treat phases as one fluid incorporating interface capture algorithms such as widely used Volume of Fluid (VOF) (Renardy & Renardy 2002; Hoang et al. 2013), level set (Sussman & Fatemi 1999; Hysing 2006), phase field methods (Liu & Shen 2003) and their refinements. An extra equation for the parameter identifying each phase is introduced and needs to be solved at each time step to update the parameter, which can degrade the computational efficiency to some extents. The interface is usually of finite thickness spanning over several meshes in fixed mesh simulation. Even though the interface may be represented sharply with either sub-grid estimation or level-set algorithm, fluid properties are smoothly transferred across the interface. The incorporation of the continuous surface force (CSF) (Brackbill et al. 1992) and later proposed continuous surface stress (CSS) methods (Raeini et al. 2012) enables the fixed mesh models to deal with surface/interface tension issues by simply adding a volumetric force term in the momentum equation. However, similar to the smooth transfer of fluid properties, those interface force models can result in the loss of the sharp force jump at the interface. Another group of meshed methods is based on adaptive mesh keeping the interface on the

mesh face (Muzaferija & Peri´c 1997; Tukovic & Jasak 2012). Rather than smearing the interface over meshes, these methods maintain the sharp interface by updating the meshes at the each time step and the interface conditions are exactly implemented maintaining both the discontinuity of fluid properties and interface forces. The main drawback of these methods is that the complex mesh reconstruction and the requirement for iterations on mesh adaptation have to be performed and are computationally expensive. These methods also encounter difficulties in dealing with topology change of the interface.

Meshless methods based on Lagrangian moving particles have recently attracted strong interest as the next generation of computational methods. It is considered to be superior to the conventional mesh based method for many applications as adopting a set of arbitrary distributed particles without using any mesh connecting these particles. Much effort is concentrated on problems for which the conventional mesh based methods are difficult to apply, such as deformable boundary, moving interface, large deformation which would involve complex mesh generation and mesh adaptively, especially for multiphase flows. For the flow type considered here, one crucial objective is the precise interface representation, which in the meshless method is inherently solved by simply tracking particles whose fluid properties are initially allocated and maintained. Even large deformations or topological changes can be well handled by flexible moving particles. In the meshless methods group, the smoothed particle hydrodynamics (SPH) is one of the matured methods (Colagrossi & Landrini 2003; Hu & Adams 2007; Szewc et al. 2015). In SPH, the multiplication of an arbitrary function and a smoothing kernel function is integrated forming the integral representation of the function which is then approximated by summing up the values of neighbouring particles. As initially developed for compressible flows (Valizadeh & Monaghan 2015; Das & Das 2011), pressure is explicitly expressed by the equation of state. Despite its easy programming and unrequired boundary treatment, problems of small time steps and pressure noise are often encountered. Incompressible SPH (Shao 2012; Zainali et al. 2013) is then proposed to overcome those two problems, but the second derivative approximation of the unknown pressure arises in solving pressure Poisson equation which is a main hurdle to achieve accuracy. This issue of solving Poisson equation with second derivative approximation is shared by other meshless methods such as meshless particle semi-implicit (MPS) method (Gotoh & Fredsøe 2000; Khayyer & Gotoh 2010) whose approximation of a function on the discrete particle is interpolated by surrounding particles using inverse distance weighting usually with the Shepard weight function. The multiphase flow solver has been developed for both SPH and MPS methods, most of which follow one fluid approach as

adopted in the fixed mesh group thus losing the sharp discontinuity of fluid properties and forces at the interface.

The Meshless Local Petro-Galerkin (MLPG) method is proposed to deal with Laplace and Poisson equations based on the local symmetric weak form and the moving least squares approximation (Atluri & Zhu 1998). The efficiency and accuracy of the MLPG method in solving Poisson's equation is significantly improved by introducing of the Rankine source solution as the test function (MLPG_R) (Ma 2005b) which omits the approximation of any derivatives of the unknown function. Although the MLPG_R has been developed in the recent decade and applied to various situations on wave structure interactions (Ma & Zhou 2009; Sriram & Ma 2012), so far there is no work that is devoted to the development of multiphase simulation using the MLPG_R method. Combining the flexible nature of meshless method in multiphase simulation and the advantages of the MLPG_R in solving Poisson equation, this study aims to extend the MLPG_R method to solve multiphase problems through implementing innovative interface conditions to cover wide range of fluid density ratios, to consider interface force and viscous effects dominant situations and to precisely capture interface particles for both non-breaking and breaking situations.

1.3 Thesis overview

The thesis starts with the introduction chapter which briefly defines the multiphase flow and the scope of this study, outlines the general analysis approaches adopted and states the aim and objectives of the study. In Chapter 2, a review of Navier-Stokes equations based numerical approaches for multiphase flow including a section for interface tension models are presented. Chapter 3 formulates the multiphase MLPG_R equations implementing interface conditions which restrict the simulation to cases with negligible interface tension and small viscous effects. To precisely identify the interface particles where the interface condition exists Chapter 4 develops a new interface identification technique based on the absolute density gradient. Chapter 5 addresses an issue concerning the determination of the discretised pressure and provides a practical and efficient approach. Combining approaches proposed in Chapter 3 to 5, a systematic validation is carried out based through a number of case studies in Chapter 6. Chapter 7 extends the interface conditions by including the effects of interface tension and high viscosity and validation tests are then performed considering flows affected by high viscous and

interface tension forces as provided in Chapter 8. Chapter 9 summarizes the study and recommends future works on simulating multiphase flows with MLPG_R method.

In this thesis, a novel multiphase flow model is proposed to simulate 2D flows of two immiscible fluids with following major achievements

- Extension of Meshless Local Petrov-Galerkin method with Rakine source solution (MLPG_R) from a single-phase model to a two-phase model
- Maintaining sharp discontinuity of fluid properties at the interface
- Applicable for fluids (either inviscid or viscous) with a wide range of density ration (1.01 to 1000) and able to consider interface tension
- Applicable for flows with large deformations and breakings

In developing this multiphase model, to achieve an effective and robust multiphase model several innovations are implemented in following aspects

- Coupling phases by an explicit pressure equation for interface particles based on newly derived interface conditions considering
    1. Jump of pressure gradient over density due to discontinuous pressure derivative in tangential direction over density, in addition with continuous pressure for fluids with small or zero viscosity and zero interface tension
    2. Modified jump of pressure gradient over density due to significant viscous effect and jump of pressure due to interface tension and normal shear stress for fluids with large viscosity and non-zero interface tension
- New technique of identifying interface particles
- New algorithms for solving algebraic pressure equation

Chapter 2 Literature Review

In fully nonlinear multiphase modelling, the numerical methods are generally based on the Navier-Stokes equations, which can be split into two groups: mesh-based and meshless methods. The former is based on Eulerian or Eulerian-Lagrangian formulation with meshes fixed or moved to adapt to the interface in which the flux of mass, momentum and energy across mesh cells are simulated. The latter is based on a Lagrangian formulation using a set of arbitrary distributed particles without any mesh that provides the connectivity of those particles. The meshless methods are more adaptive and robust for problems with free surface, deformable boundary, moving interface and large deformation which the conventional mesh based methods are difficulty to apply. In this chapter both methods dealing with multiphase flow especially for interface treatments will be reviewed but with focus on the recently popular meshless methods including methods description and their treatment of the interface. In addition, the interface tension models will also be reviewed as it is critical in simulating interface tension force dominated cases.

2.1 Mesh based methods

2.1.1 Methods based on Arbitrary Lagrangian-Eulerian formulation

The moving mesh methods use a set of separate boundary-fitted moving meshes for each phase. They represent the interface by the computational boundary, allowing accurate calculation of surface/interface tension force and direct implementation of the kinematic and dynamic conditions at the interface without any smoothing of fluid phase properties.

A mesh based method using the Arbitrary Lagrangian-Eulerian formulation was introduced for free surface flow (Muzaferija & Peri´c 1997). In this method unstructured meshes are moved to keep the grid nodes at the surface. The movement of the surface grid nodes is controlled by giving prescribed pressure on the surface and satisfying the kinematic condition and the space conservation law. But the surface tension and viscosity were neglected in this publication. The method was extended by Tukovic & Jasak (2012) and followed by Dieter-Kissling et al. (2014) to simulate interfacial flows considering interface tension and viscosity. In their development, the computational mesh consisted of two separate parts with each covering one of considered fluid phases. The two mesh parts were in contact over the two geometrically equal surfaces at the boundary of each phase (the interface) which is defined by a set of mesh faces. The mesh faces representing the interface are moved with the velocity computed from the interface conditions of continuous velocity, the balance of tangential stress and the balance of normal stress. The displacement of the interface mesh is used as a boundary condition for the solution of the mesh motion problem. With the current interface shape, pressure and velocity boundary conditions are then updated to solve each phase separately in a PISO iteration loop. The net mass fluxes through the faces at the interface were then checked if the residual levels satisfy the prescribed accuracy and if the procedure should return to interface movement to do the outer iterations.

Despite the exact interface representing and interface condition implementing, the interface mesh adapting and reconstruction are complex and time consuming. This method also encounters difficulties in dealing with changes in interface topology such as overturning or breaking waves.

2.1.2 Methods based on Eulerian formulation

Different from the approach described above in which the governing equations were arbitrary Lagragian-Eulerian formulated and were able to be written separately for each phase with jump conditions imposed to couple the solutions at the fluid interface, on the fixed meshes with Eulerian formulation, one set of equations for the whole flow domain occupied by various phases is usually solved. In one set of equations, the various phases are treated as one fluid with variable material properties that change abruptly at the interface. In this way, two issues need to be considered. One is to identify different fluids and to descript the interface evolution; the other is to consider the coupling of those interface evolution algorithms with mass and momentum equations. In this section, details of both issues will be reviewed.

Identifying the different fluids and the description of the interface evolution are generally done by using a marker function that takes different values in the different fluids. As the fluids move, and the interface between fluids changes locations, the marker function must be updated. Updating the marker function accurately is critical for the success of simulations of multiphase flows. The following section will list various marker functions that have been developed.

Volume of fluid (VOF) is a popular interface capturing method performs well in mass conservation. The cells fully occupied by one fluid are marked as 1 and others fully occupied by the other fluid are marked as zero. If there is no phase change, the value of the marker function (f) for the fluid material should be constant along its trajectory and can be expressed as

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \boldsymbol{u} \cdot \nabla f = 0 \tag{2.1}$$

during interface evolution. Combining the divergence free condition given as

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.2}$$

and integrating Eq. (2.1) over the a mesh cell with a volume of V_c and a boundary of $\partial \mathrm{V_c}$, it yields

$$\frac{\partial \alpha_V}{\partial t} + \frac{1}{V_c} \iint_{\partial V_c} \left(\boldsymbol{n}_{\partial V_c} \cdot \boldsymbol{u} \right) f dS = 0, \tag{2.3}$$

where $\alpha_V = \frac{1}{V_c} \iiint_{V_c} f dV$ with $0 \leq \alpha_V \leq 1$ is the volume fraction and $\mathbf{n}_{\partial \mathrm{V_c}}$ is the unit normal vector of cell boundary ∂V_c .

For cells with α value varying from 0 to 1, the interface can be reconstructed for geometric representation. Early work of VOF developed piece wise constant approach (Noh & Woodward 1976; Hirt & Nichols 1981) and recently further improved by Yokoi (2007). Piecewise linear approach (Youngs 1982; Rudman 1997; Aulisa et al. 2007) is the state of the art in which the interface in a cell is assumed to be a line in 2D or a plain in 3D. The unit normal of the piecewise interface in a cell is obtained from the values of α of neighbouring cells. The drawback of this approach is that the interface segments are still not continuous between neighbouring cells. More recently, piecewise parabolic (Price et al. 2001; Diwakar et al. 2009) and piecewise cubic spline (López et al. 2004) approaches have been proposed and the scheme developed by Diwakar et al. (2009) provided

continuous interface segments between cell boundaries. The VOF is also developed without interface reconstruction to relieve its complexity and the interface is located on the transition region which may cover several cells (Zalesak 1979; Muzaferija & Peri´c 1999; Xiao et al. 2005). The drawback of this scheme is the smearing of the interface and consequently numerical diffusions. Weller (2008) introduced an artificial compression term to the Eq. (2.1) to sharpen the interface and consequently to reduce the diffusions.

Level set (LS) method developed by Osher & Sethian (1988) is also a general technique to capture the interface. In this method, the indicator is a smoothed signed distance from the interface as shown in Eq. (2.1) and zero level set $f(\boldsymbol{x}, t) = 0$ gives the location of the interface. One advantage of LS method is that the interface is sharply represented and complex interfacial shapes can be handled. However, severe errors arise in mass conservation and also in interface normal vector and curvature. To maintain the level set f smooth throughout the entire simulation and to improve the mass conservation, different schemes of reinitializations for the signed distance function (Adalsteinsson & Sethian 1995; Sethian & Smereka 2003; Sussman & Fatemi 1999; Min 2010) were introduced. Even with frequent reinitializations, mass loss also happens in long time simulations. Such mass loss is corrected by global (Zhang et al. 2010) and local (Ausas et al. 2011) mass correction steps.

Other interface advection methods are also developed such as conservative level-set (Olsson & Kreiss 2005) combining VOF and LS to achieve better mass conservation of pure LS method and front tracking method in which the interface is tracked by Lagrangian marker points advected by the velocity determined from velocity field on the Eulerian grid. In the Phase-field approach (He & Kasagi 2007; Zhou et al. 2010) the interface is assumed to be of a finite thickness and the fluid dynamic is modelled by the Cahn-Hilliard equation which gives a physically realistic scalar field describing the concentration of one of the fluid components. The basic idea of this method is that an order parameter is adopted to characterize the different phases which satisfies the Cahn-Hilliard equations and is coupled to the Navier-Stokes equations.

As the Navier-Stokes equation is solved on fixed mesh and the interface is represented by indicator functions, the jump conditions at the interface are implicitly implemented and combined into the following single-field equations which are valid for the entire computational domain

$$\nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{2.4}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \mu(\nabla \otimes \boldsymbol{u} + (\nabla \otimes \boldsymbol{u})^T) + \rho \boldsymbol{g} + \boldsymbol{f}_\sigma, \tag{2.5}$$

where $\rho(\boldsymbol{x}, t)$ and $\mu(\boldsymbol{x}, t)$ are constant in each phase and discontinuous at the interface. The surface tension force locally exists at the interface and can be expressed by Dirac delta function δ_Γ with support on Γ

$$\boldsymbol{f}_\sigma = (\sigma \kappa \boldsymbol{n}_\Gamma + \nabla_\Gamma \sigma)\delta_\Gamma. \tag{2.6}$$

According to Eq. (2.4) and Eq. (2.5), density and viscosity fields and the surface tension term are determined based on the phase distribution. The surface tension issue will be discussed in section 2.3. In VOF methods the density and viscosity depend on the volume fraction α_V and the density is calculated as

$$\rho = \alpha_V \rho_1 + (1 - \alpha_V)\rho_2. \tag{2.7}$$

For the viscosity, it can be computed by arithmetic mean as

$$\mu = \alpha_V \mu_1 + (1 - \alpha_V)\mu_2 \tag{2.8}$$

and also by harmonic mean as

$$\frac{1}{\mu} = \frac{\alpha_V}{\mu_1} + \frac{1 - \alpha_V}{\mu_2}. \tag{2.9}$$

Thus, the fluid properties are computed as averages weighted by the volume fraction. Even with level-set approach in which the interface is sharply represented, the sudden jump of fluid property can have an unfavourable effect on numerical stability. To avoid it, a common approach is to set a thickness of three meshes for the interface (Sussman & Fatemi 1999; van der Pijl et al. 2005) within which the density and viscosity are smoothly transferred. Therefore, in Euler mesh, no matter the interface is sharp or smeared represented, the fluid properties are commonly smeared.

Phase-field method also represents the interface by a transition region of small but finite width which is based on models of fluid free energy. When coupling with Navier-Stokes Equations, the variable density as well as viscosity of the mixture in the fluid property transition zone is determined by mass concentration (Lowengrub & Truskinovsky 1998; Guo & Lin 2015) or alternatively the volume fraction (Liu & Shen 2003) which are also respectively used to relate the different velocities of the mixture to form the mass-

averaged velocity and volume-averaged velocity. The physical meaning of the phase-field method also enables application to many physical states of miscible, immiscible and partially miscible (Anderson et al. 1998).

2.2 Meshless methods

There are also a number of attempts to adopt meshless methods in which a finite number of discrete particles are employed to record movement of the system. Each particle can be associated with one discrete physical object, or represent a computational node in the continuum problem domain. Field variables such as mass, momentum, energy and position are possessed by each particle and the evolution of the system is determined by the conservation of mass, momentum and energy. Most meshless methods are based on the Lagrangian formulation in which particles move in a Lagrangian frame according to the internal interactions with neighbouring particles and external forces, and the convection terms do not need to be dealt with. Therefore, time history of field variables on the material can be naturally traced. For multiphase flows, unlike mesh-based methods, the interface in meshless methods is traced by Lagrangian moving particles without the need to use any marker functions. Without connection between particles, the treatment of large deformation, fragmentation and coalescence of the interface or surface is much easier. Typical meshless methods in continuum level, also known as macroscopic level, include Smooth Particle Hydrodynamics (SPH), Moving Particle Semi-implicit (MPS), and Meshless Local Petro-Galerkin (MLPG) method. The former two methods have been developed to solve multiphase flows by many researchers and the discretization of the Navier-Stokes equations and the interface treatment will be described in the following sections. MLPG has been applied in single phase flows and based on which the extension to multiphase flows with other associated techniques have been developed during this study. In this review, the formulation of weak function derived from Navier-Stokes equations and discretization techniques for the MLPG method will be briefly introduced.

2.2.1 Smooth Particle Hydrodynamics (SPH) method

SPH was first proposed by Lucy (1977) to solve astrophysical problems governed by equations of the classical Newtonian hydrodynamics. Various applications simulated were mainly focused on fluid dynamics related areas, including gravity currents (Monaghan 1994), flow through porous media (Morris et al. 1999; Holmes et al. 2011), shock

simulations (Morris & Monaghan 1997), heat transfer and mass flows (Cleary et al. 1998), underwater explosion (Swegle & Attaway 1995) etc.

The basic formulation of SPH consists of integral representation of an arbitrary function and particle approximation. With a smooth kernel function $W(\boldsymbol{x}-\boldsymbol{x}', h)$, the integral representation of function $f(\boldsymbol{x})$ is approximated by

$$\langle f(\boldsymbol{x})\rangle = \int_{\Omega} f(\boldsymbol{x}')\, W(\boldsymbol{x}-\boldsymbol{x}', h) d\boldsymbol{x}', \tag{2.10}$$

where h is the smoothing length defining the support domain of the kernel function, $\boldsymbol{x}$ and $\boldsymbol{x}'$ are position vectors of target and neighbouring particles. By doing particle approximation, the continuous integral representations as Eq. (2.10) can be discretised to be a summation over particles in the support domain (Monaghan 1994) and expressed as

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\boldsymbol{x}_j)\, W(\boldsymbol{x}-\boldsymbol{x}_j, h), \tag{2.11}$$

where N is the total number of particles in support domain, m_j and ρ_j are mass and density of particle j located at $\boldsymbol{x}_j$. For the spacial derivative, conventional expression (Monaghan 1994) can be expressed as

$$\nabla f(\boldsymbol{x}) = \sum_{j=1}^{N} f(\boldsymbol{x}_j) \frac{m_j}{\rho_j} \nabla W(\boldsymbol{x}-\boldsymbol{x}_j, h), \tag{2.12}$$

where $\nabla W(\boldsymbol{x}-\boldsymbol{x}_j, h)$ denotes the kernel gradient. To improve the accuracy, the following symmetric gradient form

$$\nabla f(\boldsymbol{x}) = \rho \sum_{j=1}^{N} m_j \left(\frac{f(\boldsymbol{x})}{\rho^2} + \frac{f(\boldsymbol{x}_j)}{\rho_j^2} \right) \nabla W(\boldsymbol{x}-\boldsymbol{x}_j, h) \tag{2.13}$$

is also widely used (Liu & Liu 2003; Monaghan 2005). One can see that in both Eq. (2.12) and Eq. (2.13) that the gradient formulations are only based on the gradient of the kernel function which can be analytically derived and consumes little computational time. So far, with the discretizations of the arbitrary function (Eq. (2.11)) and the gradient of the function (Eq. (2.12) or Eq. (2.13)), the Navier-Stokes equations can be solved by the weakly compressible SPH (WCSPH) (Antuono et al. 2012) in which the pressure is explicitly expressed by relating it to the density and the speed of sound. As pointed by Rafiee et al. (2012) and Lee et al. (2008), the advantages of WCSPH are that there is no

pressure boundary problem and its easy programming. However the drawbacks are also obvious as small time steps are required due to the sound speed adoption in the equation of state and spurious pressure oscillations arise during simulation. To obtain more stable results, artificial viscosity force is added (Colagrossi & Landrini 2003; Monaghan 2005).

Incompressible SPH (ISPH) has also been widely adopted in the field of water wave dynamics (Shao & Lo 2003; Lind et al. 2012; Liu et al. 2013; Gotoh et al. 2014). In ISPH, the pressure Poisson equation is formed by projecting the intermediate velocity onto a divergence free space by prediction-correction time marching procedure. The prediction step is to evaluate intermediate velocity $\boldsymbol{u}^*$ and position $\boldsymbol{r}^*$ without considering the pressure by

$$\boldsymbol{u}^* = \boldsymbol{u}^n + [\nabla \cdot \mu(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) + \rho \boldsymbol{g}]dt, \tag{2.14}$$

$$\boldsymbol{r}^* = \boldsymbol{r}^n + \boldsymbol{u}^* dt. \tag{2.15}$$

Thus the correction velocity $\boldsymbol{u}'$can be expressed as

$$\boldsymbol{u}' = -\frac{dt}{\rho}\nabla p^{n+1} \tag{2.16}$$

and the velocity at the next time step becomes

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{dt}{\rho}\nabla p^{n+1}. \tag{2.17}$$

Due to the invariability of density in an incompressible fluid, $\nabla \cdot \boldsymbol{u}^{n+1}$ is enforced to be zero yielding the pressure Poisson equation as follows

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p^{n+1}\right) - \frac{1}{dt}\nabla \cdot \boldsymbol{u}^* = 0. \tag{2.18}$$

In Eq. (2.18), the source term is based on the divergence of the intermediate velocity (Liu et al. 2013). The source term associated with variation of particle densities is also adopted in ISPH (Shao & Lo 2003) which is similar to the MPS method (reviewed in section 2.2.2). With Eq. (2.18), the challenge is to accurately approximate the second order derivative of pressure when discretizing the Poisson equation. Different approximation schemes have been proposed and the frequently adopted one (Shao & Lo 2003; Liu et al. 2013) is that

$$\nabla \cdot \left(\frac{1}{\rho}\nabla P\right)_i = \sum_{j=1}^{N} m_j \frac{8}{\left(\rho_i + \rho_j\right)^2} \frac{P_{ij}\boldsymbol{r}_{ij} \cdot \nabla W\left(\boldsymbol{x_i} - \boldsymbol{x}_j, h\right)}{\left|\boldsymbol{r}_{ij}\right|^2 + \eta^2}, \tag{2.19}$$

where $P_{ij} = P_i - P_j$, $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ and η is a small number introduced to keep the denominator non-zero and usually equals to $0.1h$. This symmetric formulation corresponds to a symmetric and positive definite coefficient matrix of the linear equation which can be efficiently solved. To avoid large errors in discretizing the second derivative, particularly when particles are distributed in a disorderly manner as in violent water wave problems, Fatehi & Manzari (2011) proposed several forms of a higher order scheme for Laplace operator discretization. By comparing WCSPH and ISPH (Lee et al. 2008), larger time steps could be adopted (50 times larger in one of their cases) and higher accuracy was obtained by ISPH for the same particle number. For the disadvantages, apart from solving boundary value problems, directly approximate second order derivative for disorder particle distribution is still a significant source of inaccuracy even with diversely developed schemes.

Based on above SPH theories and techniques for single phase flow, SPH has also been extended to model multiphase flows by properly treating the interface where the fluid properties are discontinuous. Similar to the single-phase flow, the multiphase SPH can be also sorted into two categories: weakly compressible SPH (WCSPH) and incompressible SPH (ISPH). In the WCSPH branch, Monaghan & Kocharyan (1995) started to adopt the WCSPH to simulate multiphase flow for the mixing dust-fluid flow, followed by gravity currents simulation (Monaghan et al. 1999). Both applications are limited in cases with small density differences as pointed out by (Hu & Adams 2006) due to their implicit assumption that the density gradient is much smaller than that of the smoothing kernel. To extend the simulation to large density ratios, Colagrossi & Landrini (2003) modified the spatial derivative approximation to diminish the large density difference effects across the interface. As applied in the non-conservative density evolution scheme, a mass conservation problem may accumulate in long time simulations even though the periodic density re-initialization of the density field was applied to relieve the problem to some extent. To ensure stability of the interface, Colagrossi & Landrini (2003) also applied a large unphysical surface tension on the low-density phase and applied XSPH correction on the momentum equation, which can be regarded as velocity smoothing or addition of a large viscous term. In Hu & Adams (2006), a conservative density evolution scheme was proposed in which the neighbouring particles only contributed to the volume rather than the mass. With this scheme and the newly proposed particle-averaged spatial derivative approximation, the density discontinuity at the interface was satisfied automatically and

the derivatives of the discontinuity were properly handled. The viscosity discontinuity was also dealt with by proposing a viscous term ensuring the continuity of velocity and shear stress at the interface even though both conditions were applied at the midpoint between each pair of particles belonging to different phases which was not the real interface. Cases with free surface and complex interface were not tested in their work. Other artificial techniques were applied to maintain the stability and sharp interface, such as repulsive forces between particles of different phases (Grenier et al. 2009) and the restricting of the degree of freedom to move the interface particles (Monaghan & Rafiee 2013) which was not applicable to deal with violent multiphase flows (i.e. violent water-air sloshing). Without velocity smoothing scheme or artificial surface tension, Chen et al. (2015) simulated multiphase flows with complex interface and large density ratios by ensuring pressure and space continuity at the interface. In summary, the WCSPH models successfully simulate multiphase flows with interfaces transferring fluid properties smoothly and maintain sharp discontinuities for fluids with both high and low density ratios. However, two issues are often mentioned for the WCSPH model. One is the pressure oscillations especially near the interface due to the adoption of the equation of state relating pressure to the density. To remedy the problems, various attempts have been made, such as the summation of the particle-averaged pressure (Ritchie & Thomas 2001), the addition of artificial surface tension (Colagrossi & Landrini 2003), improvement of spatial derivatives (Hu & Adams 2006; Chen et al. 2015) and addition of repulsive force between different fluids (Monaghan & Rafiee 2013). Another issue is that the time step is required to be normally very small in the CSPH modelling, which is considered as a main factor leading to its computational inefficiency (Hu & Adams 2007). In addition, the speed of sound adopted in low density fluid (i.e. gas) is even higher than in high density fluid (i.e. water) (which is unphysical) to ensure the density fluctuations are sufficiently low (Colagrossi 2005). Consequently, the high speed of sound for low density fluid requires even smaller time steps (Colagrossi & Landrini 2003).

Comparatively, ISPH can use relatively larger time steps without considering the speed of sound and lead to better results as pressure is implicitly solved with less computational time (Hu & Adams 2007; Zheng et al. 2014). Hu and Adams (2007, 2009) developed multiphase ISPH and proposed a scheme which can ensure the discontinuity of densities across the interface to investigate multiphase flows with relatively high density ratios. They only applied their methods to simple cases without extremely violent flow or flows without surface. Shao (2012) compared coupled ISPH in which the velocity prediction step and Poisson equation solving step were processed with mixed phase particles near the

interface and decoupled ISPH in which the pressure continuity condition was applied on the interface. It was found that both methods had comparable performance on the low density ratio of 1.001 and the decoupled ISPH worked better than coupled ISPH for high relative density ratio of 1.3. Very recently, Lind et al (2015) proposed a hybrid model for air-water flow in which the air is dealt with by the CSPH while the water is treated by the ISPH. This hybrid model satisfies the requirement of air compressibility in wave slam impacts as air ejection velocity prior to impact can approach the speed of sound. At the interface, the compressible phase provided pressure boundary condition on the incompressible phase ensuring the pressure continuity and providing Dirichlet boundary condition for the solution of the Poisson equation. On the contrary, the incompressible phase provides velocity boundary conditions for the compressible phase. However, as the definition of the interface for each phase was not mentioned, it was not clear that on which particles those two conditions were applied.

As pointed out above, the ISPH needs to solve Poisson's equation for pressure. This implies that two issues have to be dealt with. One issue is related to what and how the boundary conditions are applied on the interface. Hu and Adams (2007, 2009) suggested ensuring the continuity of pressure and the ratio of pressure gradient to density. The two conditions were imposed in their work by assuming that the middle point between each of a pair of particles is on the interface. As the pressure gradient is determined averagely by using many pairs of particles in the range of an influence domain of a concerned point, the interface in their formulation was actually a layer with a thickness in an order of the influence domain size, about double distance of two particles. In the coupled ISPH of Shao (2012), the pressure of different phase is not distinguished. In this approach, the interface is actually smeared, which was perhaps one of reasons why the coupled model did not work as well as the decoupled model. In the decoupled ISPH (Shao 2012), only pressure is ensured to be continuous but not the velocity at the interface. The second issues related to the ISPH are about how to discretise the Laplace operator involved in the Poisson's equations for pressure. Almost all the ISPH models chose to directly approximate the second order derivatives in the Laplace operator. Many schemes have been proposed as reviewed by Zheng et al. (2014). However, the errors of the schemes can be more than 30% in many points when they are applied to estimate the values of Laplace operator on simple functions if the points are irregularly distributed. Furthermore, the inaccuracy is deteriorated with the increase of irregularity in the particle distribution (Zheng et al. 2014). As the distribution of particles in the violent motion is always

irregular, the concern does exist with direct approximation to the second order derivatives of the Laplace operator.

2.2.2 Moving Particle Semi-implicit (MPS) method

The moving particle semi-implicit (MPS) method (Koshizuka & Oka 1996) for single phase flow is analogous to the SPH method providing approximations to differential equations on the basis of integral interpolation. However, there are three typical differences between MPS and SPH. The first is that the original MPS applies simplified differential operator models based on local weighted averaging process as shown in Eq.(2.23) and Eq. (2.27) rather than taking the gradient of a kernel function as in SPH. A further difference is that MPS uses a semi-implicit prediction-correction process to solve the Navier-Stokes equation by treating the fluid as fully incompressible. This prediction-correction project procedure is different from conventional WCSPH but is similar to the ISPH method. Finally, the particle number density is introduced in the MPS method to form the source term of the Poisson equation as shown in Eq.(2.22) rather than velocity divergence used in some of ISPH methods (Liu et al. 2013).

The prediction and correction step is similar to that in ISPH as shown in Eq. (2.14) to (2.16). The incompressibility of the MPS is ensured by satisfying the constant fluid density and consequently the particle number density as the density of each particle remains constant during simulation. n_0, n^* and n' denote particle number density at each time step, intermediate time step (after prediction) and the correction respectively which satisfy

$$n^* + n' = n_0. \tag{2.20}$$

By relating the velocity correction $\boldsymbol{u}'$ with n' through the mass conservation equation, one can obtain

$$\frac{1}{n_0}\frac{n'}{\Delta t} + \nabla \cdot \boldsymbol{u}' = 0. \tag{2.21}$$

Substituting Eq. (2.16) into Eq. (2.21) and replacing correction particle number density n' by $n_0 - n^*$, Eq. (2.21) can be rearranged as

$$\nabla^2 p = \frac{\rho}{\Delta t^2}\frac{n_0 - n^*}{n_0}. \tag{2.22}$$

In Eq. (2.22), the source term of the pressure Poisson equation are particle number densities at each time step, which is a constant from the initial setting and intermediate particle number density determined by the intermediate particle position at each time step. The Laplace operator on the LHS of Eq. (2.22) is represented by the particle interaction model (Koshizuka et al. 1998) as expressed below

$$\langle \nabla^2 \phi \rangle_i = \frac{2D_s}{n_0 \lambda} \sum_{j \neq i} (\phi_j - \phi_i)\, w(\boldsymbol{x}_j - \boldsymbol{x}_i, r_0), \tag{2.23}$$

where ϕ is an arbitrary scalar function, D_s is number of space dimensions, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are coordinates of central particle and neighbouring particle respectively, λ is a parameter based on particle distribution and given by

$$\lambda = \frac{\sum_{j \neq i} w(\boldsymbol{x}_j - \boldsymbol{x}_i, r_0) \left| \boldsymbol{x}_j - \boldsymbol{x}_i \right|^2}{\sum_{j \neq i} (\left| \boldsymbol{x}_j - \boldsymbol{x}_i \right|)}, \tag{2.24}$$

where n_0 is the constant particle number density calculated at the very beginning calculated as

$$n_0 = (n_i)_0 = \sum_{j \neq i} w(\boldsymbol{x}_j - \boldsymbol{x}_i, r_0). \tag{2.25}$$

The most common kernel function applied in MPS (Koshizuka et al. 1998) is

$$w(\boldsymbol{x}_j - \boldsymbol{x}_i, r_0) = \begin{cases} \dfrac{r_0}{\left| \boldsymbol{r}_j - \boldsymbol{r}_i \right|} - 1, & r \leq r_0 \\ 0, & r > r_0 \end{cases}, \tag{2.26}$$

where r_0 is the cut-off radius.

After solving the Eq. (2.22) velocities need to be updated by pressure gradient with the operator of

$$\langle \nabla \phi \rangle_i = \frac{D_s}{n_0} \sum_{j \neq i} \frac{(\phi_j - \phi_i)}{\left| \boldsymbol{r}_j - \boldsymbol{r}_i \right|^2} w(\boldsymbol{x}_j - \boldsymbol{x}_i, r_0)(\boldsymbol{x}_j - \boldsymbol{x}_i). \tag{2.27}$$

The original MPS method was proposed for single phase flows such as water column collapse (Koshizuka & Oka 1996). In that publication, the kernel as shown in Eq. (2.26) was first employed to maintain the incompressibility as the kernel function was infinite when particles overlapped which leaded to a stronger repulsive force between closer particles through the Poisson equation. Further development was made by Koshizuka et al.

(1998) in which a candidate list with the radius larger than the support domain was employed and from this list the particles in the support domain were picked out. The frequency of updating the candidate list based on the CPU time was discussed and it concluded that the scale of operation was reduced from N^2 to $N^{1.5}$ where N denoted the total particle number. Techniques to deal with floating bodies were also proposed by using a passively moving-solid model to describe the motion of a rigid body in a fluid. In (Gotoh et al. 2001), a sub-particle-scale (SPS) turbulence model was introduced for high Reynolds number flow simulations based on the concept of sub-grid-sacle (SGS) model in Eulerian large-eddy simulation (LES) models. Above mentioned techniques are reviewed by (Gotoh & Sakai 2006) and were applied on breaking wave over a slope (Koshizuka et al. 1998) and over different seabed geometries (Gotoh & Sakai 1999). As for the interaction with structures, MPS was extended to compute nonlinear motions of a floating body influenced by water on the deck and to transient dynamic loads with finite elastic structural displacements (Hwang et al. 2014).

Based on classic MPS gradient and Laplace approximation, several modifications were proposed. A corrected version of pressure gradient approximation (Khayyer & Gotoh 2008) guarantees the conservation of both linear and angular momentum by deriving an anti-symmetric pressure gradient term which was applied on plunging breaking waves. However it was pointed out by Tsuruta et al. (2013) that such correction was equivalent to adding an artificial repulsive force term which was dominantly excessive and failed to produce the main flow features. To solve this problem Tsuruta et al. (2013) proposed a dynamic stabilization scheme in which the repulsive force term was in accordance with particle distribution. To improve and stabilize the pressure during the simulation, a higher order Laplace approximation for pressure and viscous term was derived (Khayyer & Gotoh 2010) and showed enhanced pressure calculation on sinusoidal pressure oscillations and violent sloshing. Meanwhile, a higher order source term was derived to form a new pressure Poisson equation by using higher order time differentiation of particle number densities. The above improvements of Laplace approximation and Poisson equation were then extended to 3D MPS in (Khayyer & Gotoh 2012). The alteration of the source term was also done by (Kondo & Koshizuka 2011; Chen et al. 2014) to suppress the pressure oscillation in MPS. Tsuruta et al. (2015) introduced the space potential particles (PPS) to resolve the inconsistency in the volume conservation around free surface. The PPS are novel particles corresponding to each surface particle in the single phase flow and their locations are determined by particle number density deviations from that of inner particles.

In the fields of multiphase flow, the pioneering work was done by Gotoh & Fredsøe (2000) who developed a solid-fluid two-phase MPS method. Recently a weakly compressible MPS (Shakibaeinia & Jin 2012b) was developed introducing the equation of state to original MPS and presenting the ability to simulate multiphase flow with low density ratios (up to 5). Similar to the early stage SPH (Hu & Adams 2006) multiphase work, this modification comprised of density smoothing scheme giving a spatial averaging of density as expressed below

$$\langle \rho \rangle_i = \frac{1}{n_0} \sum_{j \neq i} \rho_j w(\boldsymbol{x}_j - \boldsymbol{x}_i, r_0). \tag{2.28}$$

This smoothed scheme for density on interfaces was also similar to mesh-based methods such as VOF (Yokoi 2007) and level-set (Olsson & Kreiss 2005) presenting an interface smoothly transferring the fluid properties. It was pointed out by Khayyer and Gotoh (2013) that the smooth scheme downgrades the accuracy of sharp variations of density at the interfaces and subsequently led to unphysical dispersions of particles near the interface representing as unphysical penetration of heavier fluid into lighter fluid in immiscible two-phase simulations (Shakibaeinia & Jin 2012b). To minimizing density diffusion and maintain a sharp interface, Khayyer & Gotoh (2013) presented a first order density smooth scheme based on Taylor series expansion expressed as

$$\langle \rho \rangle_i = \frac{1}{n_0} \sum_{j \neq i} \left(\rho_j - \frac{\partial \rho_i}{\partial x_{ij}} x_{ij} - \frac{\partial \rho_i}{\partial y_{ij}} y_{ij} \right) w(\boldsymbol{x}_j - \boldsymbol{x}_i, r_0). \tag{2.29}$$

This extension effectively kept the sharpness of special density variations and minimized the unphysical perturbations. Thus the performance of multiphase MPS was significantly enhanced for modelling flows with relatively high density ratio which reached 1000. Other efforts have been taken to maintain clear interface for high density ratio simulations. One of them is to incorporate the weak compressibility into Poisson's equation (Natsui et al. 2014) which was previously applied in single phase flow (Tanaka & Masunaga 2010) to obtain a constant fluid volume and smoother pressure distribution. When applied in multiphase flow, especially for large density ratios (e.g. gas-liquid flow), the weak compressibility improves the stability of the interface even with special averaged density function (Shakibaeinia & Jin 2012b). However, improper values of the compressive parameter led to decrease of the fluid volume. Thus optimization tests of the parameter were done in a hydrostatic case (Natsui et al. 2014) to obtain both clear interface and constant volume. Another technique to stabilise the gas-fluid interface is to setup a checker-board initial particle distribution (Natsui et al. 2014) rather than conventional

simple cubic distribution. The problem of particle interpenetration (no sharp interface) at the interface was also pointed out by (Ng et al. 2015) and was relieved by solving the pressure Poisson equation on the background mesh whose intermediate velocities were interpolated from moving particles. The densities at the interface were determined by the level-set method. In (Ng et al. 2015) particle deleting and feeding technique was applied to ensure the even particle distributing without using unphysical collision model (Shakibaeinia & Jin 2012b).

2.2.3 Meshless Local Petro-Galerkin (MLPG) method

The MLPG method was proposed by (Atluri & Zhu 1998) to deal with Laplace and Poisson equations based on the local symmetric weak form and the moving least squares approximation. The method successfully solved fracture mechanics problems (Batra & Ching 2002), beam and plate bending problems (Atluri & Zhu 2000) and three dimensional elasto-static and elasto-dynamic problems (Han & Atluri 2004b; Han & Atluri 2004a). Atluri et al. (2006) presented the MLPG mixed finite difference method (FDM) to solve the solid mechanics problems, in which the displacements, displacement gradients, and stresses were interpolated independently using identical MLS shape functions. The MLPG mixed finite difference method successfully solved various elasticity problems with complex displacement and stress solutions. For fluid dynamic problems, the steady flow around a cylinder, steady convection-diffusion flows in one and two dimensions and the incompressible flows by solving Navier-Stokes equations (Lin & Atluri 2001) were addressed. It was then developed to solve convection-diffusion flow problems (Lin & Atluri 2000).

The method is fully general in solving general boundary value problems and linear Poisson's equation is used here to demonstrate the formulation. The Poisson's equation can be written as

$$\nabla^2 f(\boldsymbol{x}) = g(\boldsymbol{x}) \qquad \boldsymbol{x} \in \boldsymbol{\Omega}, \tag{2.30}$$

where $g(\boldsymbol{x})$ is the given source and the domain $\boldsymbol{\Omega}$ is enclosed by the boundary consisting of Γ_f and Γ_q with the boundary conditions of

$$f = \overline{f} \quad \text{on } \Gamma_f, \tag{2.31}$$

$$\frac{\partial f}{\partial n} = \overline{q} \quad \text{on } \Gamma_q, \tag{2.32}$$

where $\overline{f}$ and $\overline{q}$ are the prescribed potential and normal flux, respectively , on the essential boundary Γ_f and flux boundary Γ_q. $\boldsymbol{n}$ is the normal direction pointing outward of the entire boundary. Other than Galerkin finite element or element free Galerkin methods which give global weak forms over the entire domain Ω, the MLPG employs the local weak form over a local sub-domain Ω_s which is located inside the global domain and is conveniently taken as a circle in 2D and a sphere in 3D problems. A generalized local weak form with the penalty method (Atluri & Zhu 1998; Atluri & Shen 2002) to implement the essential boundary condition can be written as

$$\int_{\Omega_s} (\nabla^2 f - g)\varphi \mathrm{d}\Omega - \alpha \int_{\Gamma_f} \left(f - \overline{f}\right)\varphi \mathrm{d}\Gamma = 0, \tag{2.33}$$

where f is the trial function, φ is the test function and Γ_f is a part of the boundary $\partial\Omega_s$ over which the essential boundary condition is specified. The second term of Eq. (2.33) vanishes when the domain Ω_s is entirely inside the global domain and there is no interaction between $\partial\Omega_s$ and Γ_f. α is a penalty parameter giving the value of $\alpha \gg 1$. By applying Gauss's theorem and choosing a test function φ vanishing over $\partial\Omega_s$ (excluding the interaction part between the global domains), the Eq. (2.33) is rearranged to a generalized local symmetric weak form expressed as

$$\begin{aligned} &\int_{\Omega_s} \nabla f \nabla \varphi d\Omega + \alpha \int_{\Gamma_f} f\varphi d\Gamma - \int_{\Gamma_f} \frac{\partial f}{\partial n}\varphi d\Gamma \\ &\qquad = \alpha \int_{\Gamma_f} \overline{f}\varphi d\Gamma + \int_{\Gamma_q} \overline{q}\varphi d\Gamma - \int_{\Omega_s} g\varphi d\Omega. \end{aligned} \tag{2.34}$$

Theoretically, the Eq. (2.30) and its boundary conditions will be satisfied as long as the summation of all local domains covers the global domain, i.e., $\cup\, \Omega_s \supset \Omega$. But Atluri & Zhu (1998) pointed out that satisfactory computational results were given even when the summation of sub-domains did not cover the global domain. For the trial function in Eq. (2.34), it requires a local interpolation or approximation to present unknown function of f. In MLPG, there are a variety of independent choices of nodal trial and test functions applying over nodal sub-domains with different sizes and shapes. These features make the MLPG method very flexible. Various methods (i.e., Moving least square (MLS), Shepard function, Partition of Unity methods (PU), Reproducing kernel particle methods (RKPM) and Radial basis functions (RBF)) of approximating a trial

function over an arbitrary domain using its value (or fictitious value) at a finite number of randomly located particles were discussed in Atluri and Shen (2002). In the following MLPG development, MLS interpolation for trial function was widely adopted (Batra & Ching 2002; Han & Atluri 2004a; Atluri et al. 2006) for its reasonable accuracy and high order continuity. By comparing six test functions it was found that the Heaviside step function was robust as the corresponding MLPG method does not involve any domain and singular integrals to generate the global stiffness matrix and only involves a regular boundary integral (Atluri & Shen 2002).

With the generalized formulation of the MLPG method, the incompressible flow problems were addressed by solving Navier-Stokes equations (Lin & Atluri 2001) and the method was first extended to simulate the nonlinear water waves by (Ma 2005a) in which after time-split procedure, the Poisson's equation was formulated as

$$\nabla^2 p - \frac{\rho}{\Delta t}\nabla \cdot \boldsymbol{u}^* = 0 \tag{2.35}$$

with the source term of velocity divergence where $\boldsymbol{u}^*$ was the intermediate velocity considering gravity and viscous force (not considered in (Ma 2005a) but capable to add), ρ was the density, Δt was the time step in simulation discretization and p was the pressure which would be solved by MLPG method. At each of the inner particles, a sub-domain was specified as a circle in two-dimensional simulations and over which the product of the Eq. (2.35) and an arbitrary test function φ was integrated leading to

$$\int_{\Omega_I} \left(\nabla^2 p - \frac{\rho}{\Delta t}\nabla \cdot \boldsymbol{u}^*\right) \varphi \mathrm{d}\Omega = 0, \tag{2.36}$$

where Ω_I is the area of the sub-domain centred at the particle I. In MLPG, the choice of test function is flexible and many variants of MLPG methods have been developed by (Atluri & Shen 2002) in which six test functions were compared numerically and the Heaviside step function was tested to be more promising than others. By employing the Heaviside step function expressed as

$$\varphi = \begin{cases} 1, & in\ \Omega_I \\ 0, & otherwise \end{cases} \tag{2.37}$$

and applying Gauss's theorem to Eq. (2.36), Ma (2005a) obtained the weak formulation for particles whose sub-domain Ω_I is entirely within the global domain (e.g., excludes the boundary particles) shown as

$$\int_{\partial\Omega_{\mathrm{I}}} \boldsymbol{n}\cdot\nabla p dS = \frac{\rho}{\Delta t}\int_{\partial\Omega_{\mathrm{I}}} \boldsymbol{n}\cdot\boldsymbol{u}^{*}\, dS. \tag{2.38}$$

Instead of the second derivative in Poisson's equation, the weak form shown in Eq. (2.38) only needs to deal with the gradient of the pressure. The unknown function (e.g., the trial function) at a specific position is approximated by a set of nodes located in its support domain and can be written as

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \Phi_j(\boldsymbol{x}) f(\boldsymbol{x}_j), \tag{2.39}$$

where N is the number of total particles that affect the central particle located at $\boldsymbol{x}$, $\Phi_j(\boldsymbol{x})$ is the interpolation function also called the shape function, which is formulated by using moving least square (MLS) method (Atluri & Zhu 1998; Ma 2005a; Han et al. 2006). And the derivative of an unknown function was found by taking direct differentiation of the shape function.

As for boundary particles typically used for surface and wall particles, the implementation of boundary conditions is altered from the original MLPG in non-linear wave simulation. Firstly, the surface particles on which zero pressure is specified (similar to essential boundary condition in (Atluri & Zhu 1998)) are solved by specifying zero pressure rather than in integral form as shown in penalty method. The reason is that the integral form works well for fixed or small displacement boundary problems while it is not suitable for the water wave problem as relatively large deformation occurs on the free surface (Ma 2005a). Similar free surface conditions by direct integration are also adopted in other meshless particle methods (Koshizuka & Oka 1996; Lee et al. 2008). Secondly, according to numerical tests in (Ma 2005a), wall particles satisfying zero normal pressure gradient condition which is similar to flux boundary condition (Atluri & Zhu 1998) are also formulated by direct approximation of the pressure gradient rather than the integration over the incomplete sub-domain.

A further development was carried out using a new form of the MLPG method which is called MLPG_R. In that method, the solution for Rankine sources rather than the Heaviside step function was taken as the test function (details in derivation are in section 3.2). Based on this test function, a weak form of governing equations was derived as

$$\left(\int_{\partial\Omega_I} \frac{\boldsymbol{n}\cdot(p\nabla\varphi)}{\rho} dS - \frac{p}{\rho}\right) = \int_{\Omega_I} \frac{\boldsymbol{u}^{*}\cdot\nabla\varphi}{dt} d\Omega, \tag{2.40}$$

which did not contain any derivatives of unknown functions and therefore made numerical discretization of the governing equation relatively easier and more efficient. This feature overcomes the problems associated with direct numerical approximation to second derivatives in existing ISPH and MPS formulation and the first derivatives in original MLPG method. Therefore, in this weak formulation, the only need is to approximate the unknown functions during discretizing the Poisson's equation. As a result, the approximation of the unknown functions in this approach is required to be only integrated. Clearly, this is an advantage over requiring the approximation of unknown functions to have second order derivatives. Zheng et al (2014) has shown that the use of this approach can significantly enhance the efficiency and robustness of numerical methods. A semi-analytical technique was also developed for the MLPG_R method (Ma 2005b) to evaluate the domain integral involved in this method, which dramatically reduces the CPU time spent on the numerical evaluation of the integral. Numerical tests showed that the MLPG_R method could be twice as fast as the MLPG method for modelling nonlinear water wave problems. Simulations of 2D freak waves, sloshing wave and various nonlinear water waves were successfully carried out by MLPG_R method (Ma 2007). Ma (2008) made another step forward in the development of the MLPG_R method for water waves in which a new meshless interpolation was suggested, which is as accurate as the moving least square method but is much more efficient, particularly for computation of gradients of unknown functions.

Taking the advantage of the meshless particle method in dealing with flows with large deformations or even breaking waves, the MLPG_R method was further developed to model the violent waves and their interactions with rigid structures for 2D cases (Ma & Zhou 2009) and 3D cases (Zhou & Ma 2010). In the above two works, a surface identification technique was proposed by incorporating three auxiliary functions into the particle number density method (PNDM). The new technique accurately identifies the free surface particles even in violent situations by additionally counting occupied quarters and specified rectangles of the support domain of the target particle. Compared to PNDM (Koshizuka & Oka 1996; Khayyer & Gotoh 2009), the accuracy of the identification is less dependent on regularity of particle distribution which is downgraded in violent situations. Ma & Zhou (2009) also adopted a source term of pressure Poisson equation incorporating the density invariant term used in the MPS (Koshizuka et al. 1998; Hwang et al. 2014; Khayyer & Gotoh 2012) into the velocity divergence in original MLPG_R method to overcome the particle distortion under violent conditions. A constant weighting of the density invariant term was obtained after numerical tests. Such incorporation of the source term was also discussed by incompressible SPH (Hu & Adams 2007; Gui et al.

2014). Hu & Adams (2007) achieved both density and velocity divergence constrains by double solving the pressure Poisson equation which increased the expense of CPU time. Alternatively, (Gui et al. 2014) combined two constrains with numerically determined weights whose variation was also given based on different flow conditions. Apart from the surface particle identification technique and density invariant introduced source term, the Newtonian viscosity was also considered by adding a viscous term in calculating the intermediate velocity in MLPG_R method (Ma & Zhou 2009). Another application called hydro-elasticity was also achieved by MLPG_R (Sriram and Ma, 2012) by solving fluid and structure dynamics separately while matching conditions on their interface by interaction. Besides, a novel approach was proposed to estimate pressure gradient when updating velocities and positions of fluid particles, leading to a relatively smoother pressure results. However, the MLPG_R method is up to now only applied to single phase flow and the extension to the simulations for multiphase flow is carried out in the following chapters.

2.3 Interface tension models

Since fluid molecules experience an uneven attractive molecular force near or at the surface and an abrupt change occurs when fluid properties change discontinuously, interface tension is an inherent characteristic of material interfaces of fluids. Thus, the effects of interface tension play important roles on many natural and industrial multiphase flow phenomena, especially when the characteristic length scales of the system are sufficiently small and become relevant compared to inertia effects. Examples can be found in the investigations of capillarity (Prosperetti 1981; Popinet 2009; Wang & Tong 2010), droplet dynamics (Hysing et al. 2009; Tripathi et al. 2015), marangoni effects (Haj-Hariri et al. 1997; Ma & Bothe 2011) and surfactant behaviour (Stone 1990; Xu et al. 2006). Numerical simulations of interface tension highly rely on the interface representation based on which the curvatures are estimated and interface tension models or interface boundary conditions are implemented. The following review is divided into two groups. The first group is that the interface tension is treated as a localized body force distributed within a transition region of finite thickness at the interface and the other is to simulate sharp interface force applied on an infinite thin interface.

In the first group, a continuous surface force (CSF) model interpreting surface tension as a continuous effect across an interface (Brackbill et al. 1992) was adopted which alleviates constrains of topological complexity of the interfaces. By satisfying the Laplace's formula

with the constant surface tension coefficient σ, a volumetric surface force was derived and shown as

$$F_{sv}(\boldsymbol{x}) = \sigma\kappa(\boldsymbol{x})\frac{\nabla\tilde{c}(\boldsymbol{x})}{[c]}, \tag{2.41}$$

where κ is the curvature, $\tilde{c}$ is the color function varying smoothly from one constant to another over the interface in the original CSF work and $[c]$ is the jump of two color constants. The normal vector and curvature were also proposed as

$$\boldsymbol{n} = \frac{\nabla\tilde{c}}{|\nabla\tilde{c}|}, \quad \kappa = -\nabla\cdot\boldsymbol{n}. \tag{2.42}$$

This model was first implemented in VOF method (Brackbill et al. 1992) and successfully applied on equilibrium and non-equilibrium rod, Rayleigh-Taylor instability, wall adhesion and low-gravity flows. It reported that the smoothness of the curvature and correspondingly the surface force was dependent upon the smoothness of the colour function in which the density was adopted as the colour function. The smoothed colour function in VOF to compute the curvature and the surface tension force was then adopted in Ubbink & Issa (1999) and Xiao et al. (2005). For interface reconstructed VOF, the colour function became discontinuous on sharp represented interface which made it difficult to accurately evaluate the first and second derivatives and also leads to inaccurate results of normal vectors and curvatures (Wörner 2012). Apart from VOF, the CSF model is also implemented in Level-set method by replacing the colour function with a regularized continuous delta function associated with the level-set function (Hysing 2006; Raessi et al. 2009). Benefitting from the smooth level-set function across the interface, the calculations of normal vector and curvature becomes much easier and accurate. But in either method, when calculating the curvature, convergent problems occur with mesh refinement (Cummins et al. 2005; Desjardins et al. 2008). As a remedy, a least square approach adopted by Desjardins et al. (2008) in the accurate conservative level set method was tested to be first order convergent and furthermore height function approach implemented in both Level-set (Owkes & Desjardins 2013) and VOF (Owkes & Desjardins 2015) methods achieved a second-order convergence. But extra computations are required to integrate the liquid volume fraction to build the height function for the concerned and neighbouring cells.

Similar to CSF, the continuous surface stress (CSS) model (Lafaurie et al. 1994; Renardy & Li 2001; Bothe et al. 2011) treats surface tension force as a divergence of a surface

stress tensor acting tangential to the interface. The conservative nature of the CSS ensures the zero net surface tension force for a closed interface which is superior to the CSF.

For meshless particle approaches, although the interface is inherently traced by the moving of particles, the CFS (Monaghan 2000; Zainali et al. 2013) and the CSS (Hu & Adams 2006; Hu & Adams 2007) are widely adopted by transferring the surface force to volumetric force and exerting it on a layer of interface region as in mesh based methods. To reflect the reality of the non-uniformly distributed surface forces for each phase, a new density-weighted colour-gradient method (Adami et al. 2010) was proposed by using a sharp-jumped colour function. But convergence problems of curvature calculation still cannot be avoided. Zhang (2010) proposed a method to locally reconstruct the free surface after the surface particle identification which was also adopted in Zhang et al. (2012) for the interface of two-phase flows. In 2D cases, an interpolation polynomial is used to fit the curve and MLS is used in 3D cases. But interpolation polynomial constructed curves may have oscillations due to irregular interface particle distribution. Those oscillations lead to inaccurate first and second derivatives and consequently inaccurate curvatures. Although the surface and interface are sharply identified and the curvatures are calculated based on the sharply reconstructed curve, the volumetric surface forces are also applied over single particle distance (Zhang 2010) and over the scale of the kernel support domain (Zhang et al. 2012) by transferring the force to the particles near the interface.

Different from volumetric surface force method, the second group treats the interface tension as a part of the interface boundary condition which is directly implemented on the sharp interface. A mesh based interface tracking method employing the Arbitrary Lagrangian Eulerian formulation (Tukovic & Jasak 2012; Dieter-Kissling et al. 2014) can be sorted into this group as the interface is sharply represented by the moving mesh faces. For surface/interface tension calculation, Quan & Schmidt (2007) employs least square parabola fitting of the mesh faces while Tukovic & Jasak (2012) use the novel force-conservative approach (Perot & Nallapati 2003) which ensures zero net surface tension force on any closed surface without requirement of complex derivatives or curve fitting. Although such moving mesh methods allow accurate calculation of surface tension force and direct implementation of the kinematic and dynamic conditions at the interface, they are restricted to moderately deformed interfaces with constant topology and are not applicable to violent situations such as wave breaking.

Chapter 3 Methods for Multiphase Flows with Small Viscosity and Zero Interface Tension

The problem discussed in this thesis involves flows containing two immiscible and incompressible fluid phases of Q_1 and Q_2 separated by the interface Γ as illustrated in Figure 3.1. In the fluid domain, particles can be sorted into three groups: those located on the interface (referred to as interface particles), those located continuously within each phase (referred to as inner particles) and those single particles sometimes break away from their continuous phase and penetrate into the other phase crossing the interface (referred to as isolated particles). Pressure and velocity will be solved for each particle based on the Navier-Stokes equation and the particle will move according to the achieved velocity in the Lagrangian way. In this chapter, the particle motion procedures will be first carried out. Then the pressure formulations of inner particles, interface particles and isolated particles will be derived based on MLPG_R algorithm and newly proposed interface conditions.

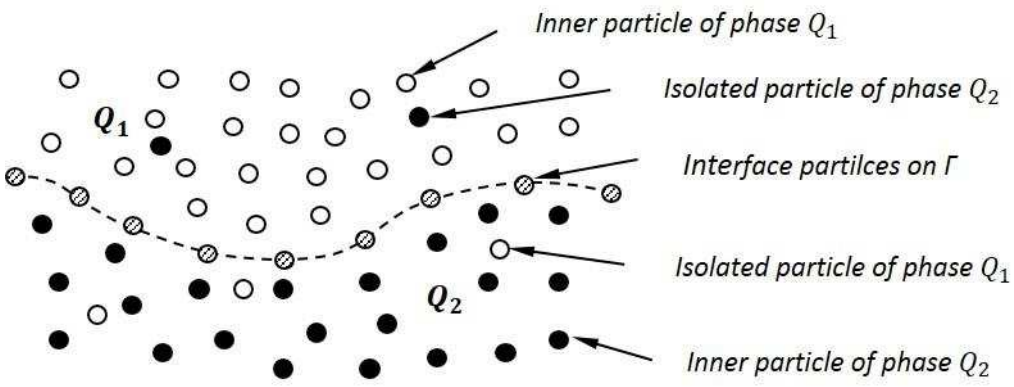


Figure 3.1: Illustration of two-phase flow consisting of phase Q_1(●) and Q_2 (○) separated by the interface Γ (- - -) represented by interface particles (⊘). There exist isolated particles surrounded by particles of the other phase.

3.1 Equations and procedure for nodes motion

The motion of particles is governed by the incompressible Navier-Stokes equations and the continuity equation in Lagrangian formulation which can be written as

$$\frac{d\boldsymbol{u}_l}{dt} = \vec{g} - \frac{1}{\rho_l}[(1-H)\nabla p_l + H\nabla p_k] + (1-H)\upsilon_l\nabla^2\boldsymbol{u}_l + H\frac{1}{\rho_l}\boldsymbol{F}_{d,l}, \tag{3.1}$$

$$\nabla\boldsymbol{u}_l = 0, \tag{3.2}$$

$$H = \begin{cases} 0, & For\ inner\ and\ interface\ particles \\ 1, & For\ isolated\ particles \end{cases},$$

where $\boldsymbol{g}$ is the gravitational acceleration, $\boldsymbol{u}$ is the fluid velocity vector, p is the pressure, ρ is the fluid density and υ is the fluid viscosity. The subscript l and k present the phase that particles belong to, where $l \in Q_1$or Q_2, $k \in Q_1$or Q_2 but $l \neq k$. It is noted that when $H = 0$, Eq. (3.1) becomes the normal momentum equation for continuous fluids, whereas $H = 1$ gives the equation of a single particle moving in the other fluid driven by the pressure gradient of the surrounding fluid, the drag force and the gravity.

The two forms of the continuity equation as shown in Eq. (3.2) are equivalent for continuous fluids but the first can be applied to isolated particles. For inner and interface particles ($H = 0$), the motions are driven by gravity, pressure gradient and viscous force as in other Lagrangian multiphase approaches (Shakibaeinia & Jin 2012b; Monaghan & Rafiee 2013) for two continuous phases. To ensure the continuity of the terms related to viscous stress on the interface, the term of $\upsilon_l\nabla^2\boldsymbol{u}_l$ in Eq. (3.1) is replaced by $(\upsilon_l\nabla^2\boldsymbol{u}_l +$

$\upsilon_k \nabla^2 \boldsymbol{u}_k)/2$ as an average for interface particles. The similar approach was also adopted by Grenier et al. (2009) and Flekkoy et al. (2000) by adopting the inter-particle averaged shear stress.

For isolated particles ($H = 1$), the total force on them is contributed by pressure gradient of the other phase, drag and gravity. Other forces such as lift force and Basset force are assumed negligible in this paper. The similar simplification was also adopted in Sokolichin and Eigenberger (1997) for the motion of air bubbles in water and in Monaghan and Kocharyan (1995) for the motion of dust particles in air. Isolated particles were also dealt with by Gotoh and Sakai (2006) in their MPS method in the occasion that single air particle was surrounded by water particles. They obtained pressure by temporarily giving water density to that air particle to consider the reaction force from the air particle to surrounding water particles while the movement of the air particle was not provided. $F_{d,l}$ in Eq. (3.1) accounts for the drag force exerted by surrounding phase on the isolated particle which was given by Odar (1964) and expressed as

$$\boldsymbol{F}_{d,l} = \frac{A_l}{2V_l} C_d \rho_k |\boldsymbol{u}_k - \boldsymbol{u}_l| (\boldsymbol{u}_k - \boldsymbol{u}_l), (l \neq k), \tag{3.3}$$

where $A_l = \pi R^2$ and $V_l = \frac{4}{3}\pi R^3$ with R being the radius of the isolated particle. C_d is the drag coefficient approximated by the standard drag curve (Clift et al. 1978; Delnoij et al. 1997):

$$C_d = \begin{cases} \frac{24}{R_e}\left(1 + 0.15{R_e}^{0.678}\right) & R_e < 1000 \\ 0.44 & R_e \geq 1000 \end{cases},$$

where $R_e = 2\rho_k R|\boldsymbol{u}_k - \boldsymbol{u}_l|/\mu_k$. In Eq. (3.1) and (3.3), the pressure p_k and velocity $\boldsymbol{u}_k$ at the centre of an isolated particle are interpolated from the continuous phase as it is assumed that a small number of isolated particles have negligible disturbance on the pressure of continuous phase based on Gotoh and Sakai (2006). Substituting Eq. (3.3) to Eq. (3.1), one can observe that there is a term of ρ_k/ρ_l, indicating the ratio of density of surrounding particles to that of the isolated particle. A high value of that term leads to a significant effect of drag force on isolated particles such as air bubbles in water. On the other hand a low value leads to a negligible effect such as on water drops in air.

On rigid wall boundaries, the following condition is satisfied

$$\boldsymbol{u_l} \cdot \boldsymbol{n} = \boldsymbol{U} \cdot \boldsymbol{n}. \tag{3.4}$$

Taking time derivatives of the velocities on both sides of Eq. (3.4) and substituting $\boldsymbol{du_l/dt}$ into Eq. (3.1) for the continuous particles, the pressure of particles on the rigid wall boundaries satisfies

$$\boldsymbol{n}\cdot\nabla p_l = \rho_l\left(\boldsymbol{n}\cdot\boldsymbol{g} - \boldsymbol{n}\cdot\dot{\boldsymbol{U}} + \upsilon_l\boldsymbol{n}\cdot\nabla^2\boldsymbol{u}_l\right), \tag{3.5}$$

where $\boldsymbol{n}$ is an unit normal vector of the wall boundary with a velocity and an acceleration of $\boldsymbol{U}$ and $\dot{\boldsymbol{U}}$, respectively.

The model is numerically solved by a time marching procedure consisting of prediction and correction steps. This procedure has been adopted and detailed in our previous papers (Ma & Zhou 2009) for single phase flow and a summary is presented below. Suppose that variables of pressure, velocity and location of each particle have been found at n-th time step ($t = t^n$) and those will be updated at (n+1)-th time step.

(a) Prediction step

The time domain is first split into steps of dt. With known velocity ($\boldsymbol{u}_l{}^n$) and particle position ($\boldsymbol{r}_l{}^n$) at n-th time step, the intermediate velocity ($\boldsymbol{u}_l{}^*$) and position ($\boldsymbol{r}_l{}^*$) are explicitly calculated according to Eq. (3.1) by considering forces excluding the term associated with the pressure and yield

$$\boldsymbol{u}_l{}^* = \boldsymbol{u}_l{}^n + \boldsymbol{g}dt + \upsilon_l\nabla^2\boldsymbol{u}_l dt, \tag{3.6}$$

$$\boldsymbol{r}_l{}^* = \boldsymbol{r}_l{}^n + \boldsymbol{u}_l{}^* dt. \tag{3.7}$$

(b) Correction step

Next by considering the pressure term, the velocity at (n+1)-th time step can be expressed as

$$\boldsymbol{u}_l{}^{n+1} = \boldsymbol{u}_l{}^* - \frac{\Delta t}{\rho_l}\nabla p_l{}^{n+1}. \tag{3.8}$$

Then by taking divergence of both sides of Eq. (3.8) and utilizing mass conservation equation (Eq. (3.2)), the pressure Poisson's equation of $p_l{}^{n+1}$ is derived as

$$\nabla^2 p_l{}^{n+1} = \frac{\rho_l}{dt}\nabla\cdot\boldsymbol{u}_l{}^*. \tag{3.9}$$

After obtaining the pressure of each particle by solving pressure Poisson's equation, the correction velocity will be estimated as

$$\boldsymbol{u}_l^{**} = -\frac{dt}{\rho_l}\nabla p_l^{n+1}. \tag{3.10}$$

Finally, the full time step velocity and position will be updated by

$$\boldsymbol{r}_l^{n+1} = \boldsymbol{u}_l^{*} + \boldsymbol{u}_l^{**}, \tag{3.11}$$

$$\boldsymbol{r}_l^{n+1} = \boldsymbol{u}_l^{n} + \boldsymbol{u}_l^{n+1} dt. \tag{3.12}$$

It should be noted that the whole procedure of Eq. (3.6) - (3.12) is only applicable for inner particles. For interface particles, the pressure is not governed by Eq. (3.9) and will be discussed in Section 3.3, though their velocities and positions are also updated by Eq. (3.10) - (3.12) once the pressure is obtained. For isolated particles, the pressure equation will be presented in Section 3.4. Their velocities are estimated by using Eq. (3.1) with setting $H = 1$ and their positions are updated by Eq. (3.12).

As shown in Figure 3.1, three types of nodes, e.g. inner nodes, interface nodes and isolated nodes, exist in the simulation with different pressure formulations. In the next three sections 3.2 - 3.4, pressure formulations will be given for each type of node. At the inner particles, the pressure is treated using the same technique as the single phase MLPG_R method (Ma and Zhou, 2009) in section 3.2 while the pressure at the interface and isolated particles is formulated by considering interface conditions in section 3.3 and 3.4 respectively.

3.2 MLPG_R formulation for the inner particles

For inner particles, the pressure is implicitly solved by Eq. (3.9) for fluid ones and by Eq. (3.5) for wall boundary ones. Following the formulation in single phase MLPG_R (Ma 2005b), for each inner fluid particle, a circular sub-domain Ω_I centred at node I is specified over which the production of Eq. (3.9) and an arbitrary test function φ is integrated, leading to

$$\int_{\Omega_I} \left(\nabla^2 p_l^{n+1} - \frac{\rho_l}{dt}\nabla \cdot \boldsymbol{u}_l^{*}\right)\varphi = 0. \tag{3.13}$$

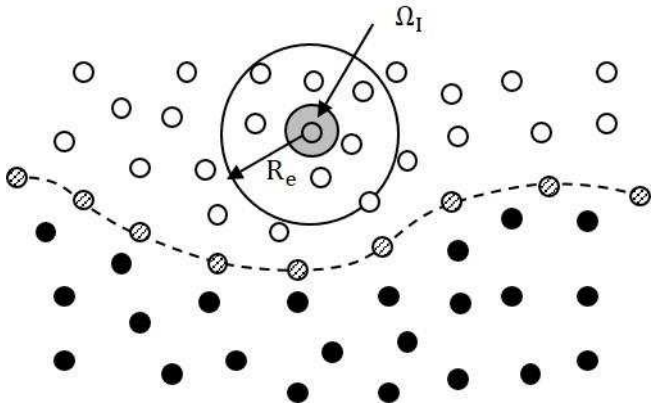


Figure 3.2: Illustration of arbitrary distributed particles, integration domain and support domain

Several options of the test function have been explored and it was found that with the Heaviside step function the domain integral over Ω_I was avoided and only boundary integrals were involved which greatly improved the effectiveness of this method (Atluri & Shen 2002). When dealing with water wave problems, the Heaviside step function was also adopted (Ma 2005a) resulting in a formulation involving boundary integral only. However, in that formulation the gradient of the unknown function exists in the integrant which not only requires much computational time but also degrades the accuracy. To further improve the weak formulation, the Rankine source solution was selected as test function (MLPG_R) to eliminate the gradient of the unknown function (Ma 2005b). This test function satisfies that $\nabla^2\varphi = 0$ within Ω_I, except for its centre and $\varphi = 0$ on $\partial\Omega_I$ which is the boundary of Ω_I. The solution can be expressed as

$$\varphi = \frac{1}{4\pi}\begin{cases} 2\,ln\left(\frac{r}{R_I}\right) & for\ two\ dimensional\ cases \\ \left(1-\frac{R_I}{r}\right) & for\ three\ dimensional\ cases \end{cases}, \tag{3.14}$$

where r is the distance between a concerned point and the centre of Ω_I and R_I is the radius of Ω_I. With this test function, Eq. (3.14) is then rearranged by adding a zero term $-p\nabla^2\varphi$ and applying Gauss's theorem, shown as

$$\begin{aligned} &\int_{\partial\Omega_I+\partial\varepsilon}[\boldsymbol{n}\cdot(\varphi\nabla p_l)-\boldsymbol{n}\cdot(p_l\nabla\varphi)]dS \\ &\qquad=\int_{\partial\Omega_I+\partial\varepsilon}\frac{\rho_l}{\Delta t}\boldsymbol{n}\cdot(\varphi\boldsymbol{u}_l^*)ds-\int_{\Omega_I}\frac{\rho_l}{\Delta t}\boldsymbol{u}_l^*\cdot\nabla\varphi d\Omega, \end{aligned} \tag{3.15}$$

where $\boldsymbol{n}$ is the unit vector normal to integration sub-domain pointing outside and $\partial\varepsilon$ is a small surface surrounding the centre of Ω_I which is a circle in 2D cases and a sphere surface in 3D cases with a radius of ε. With the introduction of $\partial\varepsilon$, the infinite value of φ at $r = 0$ is avoided and therefore it is straightforward to apply the Gauss's theorem. By taking $\varepsilon \to 0$, one can easily prove that

$$\int_{\partial\Omega_I+\partial\varepsilon} \boldsymbol{n} \cdot (\varphi \nabla p_l) dS = 0,$$

$$\int_{\partial\varepsilon} \boldsymbol{n} \cdot (p_l \nabla \varphi) dS = -p_l$$

and

$$\int_{\partial\Omega_I+\partial\varepsilon} \boldsymbol{n} \cdot (\varphi \boldsymbol{u}_l^*) dS = 0.$$

And then the final pressure equation becomes

$$\left(\int_{\partial\Omega_I} \frac{\boldsymbol{n} \cdot (p_l \nabla \varphi)}{\rho_l} dS - \frac{p_l}{\rho_l} \right) = \int_{\Omega_I} \frac{\boldsymbol{u}_l^* \cdot \nabla \varphi}{dt} d\Omega. \tag{3.16}$$

The major difference between Eq. (3.16) and Eq. (3.9) is that Eq. (3.16) does not include any derivative of the functions to be solved while Eq. (3.9) contains the second order derivatives of unknown pressure. Approximation to the unknown functions in Eq. (3.16) does not require them to have any continuous derivatives, while approximation to the unknown functions in Eq. (3.9) requires them to have finite, or at least integrable second order derivatives. Therefore, use of Eq. (3.16) for further discretisation has a great numerical advantage over using of Eq. (3.9) directly. One of the differences between MLPG_R method and MPS or ISPH lies in adopting different pressure governing equations for further discretisation. MPS or ISPH methods discretise Eq. (3.9) directly, as indicated above.

3.2.1 Pressure approximation by Moving Least Square (MLS) method

The unknown trial function p_l in weak formulation of Eq. (3.16) needs to be approximated by a set of discretised variables and is generally written as

$$p_l(\vec{x}) \approx \sum_{j=1}^{N} \phi_j(\boldsymbol{x})\hat{p}_j, \tag{3.17}$$

where N is the number of nodes affecting the value at point $\boldsymbol{x}$, e.g. the number of neighbouring nodes within the support domain with the radius of R_e as shown in Figure 3.2, and $\hat{p}_j$ is the fictitious nodal variable on the neighbouring randomly distributed particle. ϕ_j is the shape function which can be formulated in a variety of interpolation ways in meshless methods as discussed in Atluri and Shen (2002). As the Moving Least Square (MLS) method is generally considered as one of the schemes to interpolate data with a reasonable accuracy, this scheme is chosen in the MLPG_R method. Here the function of pressure $p_l(\boldsymbol{x})$ needs to be approximated with a number of scattered particles located at $\{\boldsymbol{x}_j\}, j = 1,2,\dots,N$, resulting in an approximant ${p_l}^h(\boldsymbol{x})$ of $p_l(\boldsymbol{x})$ defined by

$${p_l}^h(\boldsymbol{x}) = \boldsymbol{\psi}^T(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}), \tag{3.18}$$

where, $\boldsymbol{\psi}^T(\boldsymbol{x}) = [\psi_1, \psi_2, \dots, \psi_m]$ is a complete monomial basis of order M which can be chosen as linear:

$$\boldsymbol{\psi}^T(\boldsymbol{x}) = [1, x, y], \quad M = 3 \tag{3.19}$$

or quadratic:

$$\boldsymbol{\psi}^T(\boldsymbol{x}) = [1, x, y, x^2, xy, y^2], \quad M = 6 \tag{3.20}$$

for 2D problems, and $\boldsymbol{a}(\boldsymbol{x})$ is a vector containing coefficients with $a_m(\boldsymbol{x})$, $j = 1,2,\dots,M$ which are functions of space coordinates and determined by minimizing a weighted discrete L_2 norm defined as

$$\begin{aligned} \boldsymbol{J}(\boldsymbol{x}) &= \sum_{j=1}^{N} w_j(\boldsymbol{x})\left[\boldsymbol{\psi}^T(\boldsymbol{x}_j)\boldsymbol{a}(\boldsymbol{x}) - \hat{u}_j\right]^2 \\ &= [\boldsymbol{\Psi}\cdot\boldsymbol{a}(\boldsymbol{x}) - \hat{\boldsymbol{u}}]^T \cdot \boldsymbol{W} \cdot [\boldsymbol{\Psi}\cdot\boldsymbol{a}(\boldsymbol{x}) - \hat{\boldsymbol{u}}], \end{aligned} \tag{3.21}$$

where $w_j(\boldsymbol{x})$ is the weight function associated with the node j in the support domain of the target particle with $w_j(\boldsymbol{x}) > 0$ and is defined by

$$w_j(\vec{x}) = \begin{cases} 1 - 6\overline{r}^2 + 8\overline{r}^3 - 3\overline{r}^4 & \overline{r} = \dfrac{r_j}{R_e} = \dfrac{|\boldsymbol{x}_j - \boldsymbol{x}|}{R_e} \le 1 \\ 0 & \overline{r} = \dfrac{r_j}{R_e} = \dfrac{|\boldsymbol{x}_j - \boldsymbol{x}|}{R_e} > 1 \end{cases}, \tag{3.22}$$

where r_j is the distance from particle $\boldsymbol{x}_j$ to the target particle $\boldsymbol{x}$, R_e is the size of the support domain. The matrices $\boldsymbol{\Psi}$ and $\boldsymbol{W}$ are defined as

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}^T(\boldsymbol{x}_1) \\ \boldsymbol{\psi}^T(\boldsymbol{x}_2) \\ \cdots \\ \boldsymbol{\psi}^T(\boldsymbol{x}_N) \end{bmatrix}_{N \times M}, \tag{3.23}$$

$$\boldsymbol{W} = \begin{bmatrix} w_1(\boldsymbol{x}) & \mathbf{0} & \mathbf{0} \\ \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & w_N(\boldsymbol{x}) \end{bmatrix}_{\boldsymbol{N \times N}} \tag{3.24}$$

and

$$\hat{\boldsymbol{u}}^T = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_N] \tag{3.25}$$

where $\hat{u}_j, j = 1,2,\dots,N$ are fictitious nodal values but not the nodal values of the unknown trial function ${p_l}^h(\boldsymbol{x})$ in general. The stationary of $\boldsymbol{J}$ in Eq. (3.21) with respect to $\boldsymbol{a}(\boldsymbol{x})$ leads to the following relation between $\boldsymbol{a}(\boldsymbol{x})$ and $\hat{\boldsymbol{u}}$ as

$$\mathbf{A}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\hat{\boldsymbol{u}}, \tag{3.26}$$

where the matrix of $\mathbf{A}(\boldsymbol{x})$ and the vector of $\boldsymbol{B}(\boldsymbol{x})$ are expressed as

$$\mathbf{A}(\boldsymbol{x}) = \boldsymbol{\Psi^T W \Psi} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} w_j(\boldsymbol{x})\, \boldsymbol{\psi}(\boldsymbol{x}_j)\boldsymbol{\psi}^{\boldsymbol{T}}(\boldsymbol{x}_j), \tag{3.27}$$

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{x}) &= \boldsymbol{\Psi^T W} \\ &= [w_1(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}_1), w_2(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}_2), \dots, w_N(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}_N)\,]. \end{aligned} \tag{3.28}$$

By solving $\boldsymbol{a}(\boldsymbol{x})$ with Eq. (3.26) and substituting it into Eq. (3.18), the weight function $\phi_j(\boldsymbol{x})$ in Eq. (3.17) corresponding to particle $\boldsymbol{x}_j$ is given by

$$\phi_j(\boldsymbol{x}) = \sum_{m=1}^{M} \psi_m(\boldsymbol{x})[\mathrm{A}^{-1}(\boldsymbol{x})B(\boldsymbol{x})]_{mj}. \tag{3.29}$$

It can be seen that the MLS is well defined only when the matrix $\mathbf{A}(\boldsymbol{x})$ is non-singular which means that the number of particles within the support domain should be equal or larger than M (i.e.,$N \geq M$) for each target particle $\boldsymbol{x}$ and also requires that the nodes in the support domain should not be arranged in a special pattern such as on a straight line in 2D problems.

3.2.2 Numerical Technique for Domain Integration

The RHS of Eq. (3.16) is the domain integration usually numerically estimated by Gaussian quadrature. In this way, more than 16 Gaussian points for 2D cases and 64 points for 3D cases may be required to achieve satisfactory results and the intermediate velocity $\boldsymbol{u}_l^*$ at each Gaussian point is estimated by the MLS method. To avoid evaluating the velocities at so many points and to improve the efficiency, a semi-analytical integration technique (Ma 2005b) is adopted in which the integration domain is divided into several sub-domains and within each sub-domain the variable is assumed to vary linearly and the integration is analytically performed. This method reduces the number of points to 5 for a 2D circle with four sub-domains and 7 for a 3D sphere with 8 sub-domains. Details of the technique will be demonstrated here by a 2D case considering a circular integration domain with the size of R_I and centred at (x_0, y_0). As shown in Figure 3.3, the intermediate velocity within each sub-domain (e.g. 0-1-2) is assumed to be linear and can be expressed as

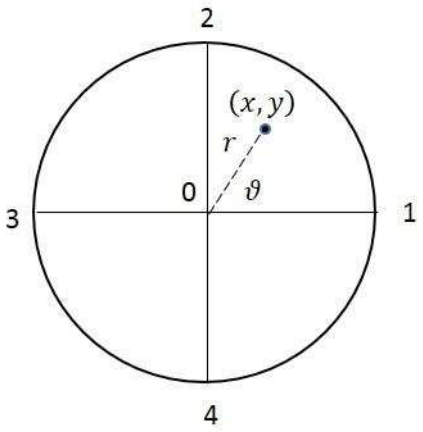


Figure 3.3: Illustration of division of an integration domain

$$u^*(x,y) = {u_0}^* + c_{ux}(x - x_0)/R_I + c_{uy}(y - y_0)/R_I, \tag{3.30a}$$

$$v^*(x,y) = {v_0}^* + c_{vx}(x - x_0)/R_I + c_{vy}(y - y_0)/R_I, \tag{3.30b}$$

where (u^*, v^*) are the intermediate velocity components at any points (x, y) within the sub-domain 0-1-2 and $({u_0}^*, {v_0}^*)$ are those at the centre of point 0. c_{ux}, c_{uy}, c_{vx} and c_{vy} are constants determined by the known velocities at point 1 and 2. Taking horizontal velocity component as an example as

$$c_{ux}(x_1 - x_0) + c_{uy}(y_1 - y_0) = ({u_1}^* - {u_0}^*)R_I,$$

$$c_{ux}(x_2 - x_0) + c_{uy}(y_2 - y_0) = ({u_2}^* - {u_0}^*)R_I,$$

gives

$$c_{ux} = \frac{({u_1}^* - {u_0}^*)(y_2 - y_0) - ({u_2}^* - {u_0}^*)(y_1 - y_0)}{(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)} R_I, \tag{3.31a}$$

$$c_{uy} = \frac{({u_2}^* - {u_0}^*)(x_1 - x_0) - ({u_1}^* - {u_0}^*)(x_2 - x_0)}{(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)} R_I, \tag{3.31b}$$

where c_{vx} and c_{vy} can be found in a similar way by considering the vertical velocity component. Thus the velocities in this sub-domain can be determined by Eq. (3.31a) and those in other sub-domains can also be determined by working out four constants with corresponding given velocities at three points. Consequently, the velocities in the whole integration domain are determined by five points (0, 1, 2, 3 and 4). Based on this, the RHS of Eq. (3.16) can be first rearranged by substituting the test function and transferring it to polar coordinate system yielding $\frac{1}{2\pi dt}\int_0^{2\pi}\int_0^{R_I} {u_{lr}}^*(r,\theta)\, dr d\theta$ where ${u_{lr}}^*$ is intermediate velocity in radial direction of one phase. The integration can be written as the summation of four sub-domains expressed as

$$\int_0^{2\pi}\int_0^{R_I} {u_{lr}}^*(r,\theta)\, dr d\theta = \sum_{i=1}^{N_i}\int_{\theta_i}^{\theta_{i+1}}\int_0^{R_I} {u_{lr}}^*(r,\theta)\, dr d\theta, \tag{3.32}$$

where $N_i = 4$ and $\theta_5 = \theta_1$. The integration over each sub-domain can be evaluated analytically using Eq. (3.30) with the integrand formulated in polar coordinate system as

$$u^*(r,\theta) = {u_0}^* + c_{ux} r cos\theta/R_I + c_{uy} r sin\theta/R_I, \tag{3.33a}$$

$$v^*(r,\theta) = {v_0}^* + c_{vx} r cos\theta / R_I + c_{vy} r sin\theta / R_I. \tag{3.33b}$$

The integrand in the radial direction can be expressed as

$$\begin{aligned} {u_{lr}}^*(r,\theta) = {u_0}^* cos\theta + {v_0}^* sin\theta + c_{ux} r cos^2\theta / R_I \\ + c_{uy} r sin\theta cos\theta / R_I + c_{vx} r cos\theta sin\theta / R_I \\ + c_{vy} r sin^2\theta / R_I. \end{aligned} \tag{3.34}$$

It is easy to show that the integration of ${u_0}^* cos\theta + {v_0}^* sin\theta$ becomes zero and can be omitted. The integration of the rest term in Eq. (3.34) over each sub-domain is given by

$$\begin{aligned} &\int_{\theta_i}^{\theta_{i+1}} \int_0^{R_I} {u_{lr}}^*(r,\theta)\, dr d\theta \\ &= \frac{R_I}{4}\big[(c_{ux} + c_{vy})(\theta_{i+1} - \theta_i) \\ &+ (c_{ux} - c_{vy})(sin\theta_{i+1} cos\theta_{i+1} - sin\theta_i cos\theta_i) \\ &+ (c_{uy} + c_{vx})(sin^2\theta_{i+1} - sin^2\theta_i)\big]. \end{aligned} \tag{3.35}$$

The whole circular domain integration is the summation of Eq. (3.35) of four sub-domains in which way the velocities at only four points are needed rather than 16 points by Gaussian quadrature integration. Although the results depend on the number of sub-domains, numerical tests show that four sub-domains are satisfactory for 2D cases (Ma 2005b).

3.2.3 Gradient estimation scheme

For inner wall particles, the rigid wall boundary condition is satisfied by Eq. (3.5) in which the gradient of unknown pressures needs to be approximated. As shown by Eq. (3.10), the velocity updating also involves gradient estimation of previously solved pressure. Thus, an accurate and efficient gradient estimation scheme is required in MLPG_R method. A simplified finite difference interpolation (SFDI) scheme was proposed by Ma (2008) for either function interpolation or gradient estimation for both known and unknown variables. This interpolation and gradient estimation scheme achieves exact accuracy for linear functions independent of particle distribution, which is better than the scheme used in MPS (Koshizuka & Oka 1996; Yoon et al. 2001) whose accuracy is highly dependent on regular particle distribution. Compared to first order

MLS which has higher accuracy than those adopted in MPS, SFDI has the same order in accuracy. While without the matrix inverse in SFDI, the efficiency is significantly improved which was demonstrated by various case studies in Ma (2008). In this section, details of the SFDI gradient estimation scheme will be presented.

An arbitrary function $f(\boldsymbol{r})$ is first expanded into a Taylor series near point $\boldsymbol{r}_i$, giving

$$f(\boldsymbol{r}) = f(\boldsymbol{r}_i) + (\nabla f)_{\boldsymbol{r}_i}(\boldsymbol{r} - \boldsymbol{r}_i) + O(|\boldsymbol{r} - \boldsymbol{r}_i|^2), \tag{3.36}$$

where $(\nabla f)_{\boldsymbol{r}_i}$ is the gradient of $f(\boldsymbol{r})$ at point $\boldsymbol{r}_i$. Applying the above expression to a set of particles j around $\boldsymbol{r}_i$ by ignoring the error term, multiplying $\frac{\boldsymbol{r}_{j,x_m} - \boldsymbol{r}_{i,x_m}}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^2} w(|\boldsymbol{r}_j - \boldsymbol{r}_i|)$ on both sides of Eq. (3.36) and taking the sum of equations at all the relevant particles yields

$$\begin{aligned}
&\sum_{j=1}^{N} [f(\boldsymbol{r}_j) - f(\boldsymbol{r}_i)] \frac{\boldsymbol{r}_{j,x_m} - \boldsymbol{r}_{i,x_m}}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^2} w(|\boldsymbol{r}_j - \boldsymbol{r}_i|) \\
&\quad = \sum_{j=1}^{N} \frac{(\boldsymbol{r}_{j,x_m} - \boldsymbol{r}_{i,x_m})^2}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^2} w(|\boldsymbol{r}_j - \boldsymbol{r}_i|)(f_{,x_m})_{\boldsymbol{r}_i} \\
&\quad + \sum_{j=1}^{N} \sum_{k=1, k\neq m}^{d} (\boldsymbol{r}_{j,x_k} - \boldsymbol{r}_{i,x_k}) \frac{\boldsymbol{r}_{j,x_m} - \boldsymbol{r}_{i,x_m}}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^2} w(|\boldsymbol{r}_j - \boldsymbol{r}_i|)(f_{,k})_{\boldsymbol{r}_i},
\end{aligned} \tag{3.37}$$

where N is the number of neighbouring particles in the support of the concerned particle i, r_{j,x_m} and r_{j,x_k} are the coordinate components of particle j, $w(|\boldsymbol{r}_j - \boldsymbol{r}_i|)$ is the weight function of particle j and $(f_{,x_m})_{\boldsymbol{r}_i}$ is the gradient component of particle $\boldsymbol{r}_i$ in the x_m direction.

Alternatively, we have

$$(f_{,x_m})_{\boldsymbol{r}_i} + \sum_{k=1, k\neq m}^{d} a_{0,mk} (f_{,k})_{\boldsymbol{r}_i} = C_{0,m} \quad (m = 1,2,\dots,d), \tag{3.38}$$

where

$$C_{0,m} = \frac{1}{n_{0,x_m}} \sum_{j=1}^{N} [f(\boldsymbol{r}_j) - f(\boldsymbol{r}_i)] \frac{\boldsymbol{r}_{j,x_m} - \boldsymbol{r}_{i,x_m}}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^2} w(|\boldsymbol{r}_j - \boldsymbol{r}_i|),$$

$$a_{0,mk} = \frac{1}{n_{0,x_m}} \sum_{j=1}^{N} \left(\boldsymbol{r}_{j,x_k} - \boldsymbol{r}_{i,x_k}\right) \frac{\boldsymbol{r}_{j,x_m} - \boldsymbol{r}_{i,x_m}}{\left|\boldsymbol{r}_j - \boldsymbol{r}_i\right|^2} w\left(\left|\boldsymbol{r}_j - \boldsymbol{r}_i\right|\right),$$

and

$$n_{0,x_m} = \sum_{j=1}^{N} \frac{\left(\boldsymbol{r}_{j,x_m} - \boldsymbol{r}_{i,x_m}\right)^2}{\left|\boldsymbol{r}_j - \boldsymbol{r}_i\right|^2} w\left(\left|\boldsymbol{r}_j - \boldsymbol{r}_i\right|\right).$$

By solving Eq.(3.38), this gradient estimation scheme can be applied in either 2D or 3D cases with $d = 2$ or $d = 3$ respectively. For 2D cases which are in the scope of this thesis, gradient components are solved and expressed as

$$\left(f_{,x}\right)_{\boldsymbol{r}_i} = \frac{C_{0,1} - a_{0,12} C_{0,2}}{1 - a_{0,12} a_{0,21}}, \tag{3.39a}$$

$$\left(f_{,y}\right)_{\boldsymbol{r}_i} = \frac{C_{0,2} - a_{0,21} C_{0,2}}{1 - a_{0,12} a_{0,21}}. \tag{3.39b}$$

So far, all the treatments are for inner and particles which follow the work in single phase flow (Ma 2005b). To extend the method to model two-phase flows, this thesis focuses on developing new implementations of interface conditions and other associated techniques in the following sections to the end of Chapter 5. The validations are carried out in Chapter 6 based on the method described in Chapter 3 to Chapter 5.

3.3 Interface conditions for fluids with small viscosity and zero interface tension

The boundary conditions at the interface of two incompressible fluids (Brackbill et al. 1992; Sussman et al. 2007) will be simplified based on the assumption of small viscosity for either fluid and zero interface tension. Without considering the effects of the interface tension, the normal pressure boundary condition between two phases (labelled l and k) is

$$p_l - p_k = \tau_{l,n} - \tau_{k,n} \quad (l, k \in Q_1 \text{or } Q_2, \text{ but } l \neq k) \tag{3.40}$$

and in the tangential direction the condition is written as

$$\tau_{l,\tau} = \tau_{k,\tau}, \tag{3.41}$$

where p_α is the pressure in fluid α for $\alpha = l, k$, $\tau_{\alpha,n}$ and $\tau_{\alpha,\tau}$ are viscous stress in fluid α in the normal and tangential direction respectively. Considering the fact that the dynamic viscosity of fluid is very small ($<10^{-3}$) for many fluids, such as water and air, the viscous stress differences at the interface between phases are neglected resulting in the automatic satisfaction of Eq. (3.41) and the simplified Eq. (3.40) written as

$$p_l = p_k. \tag{3.42}$$

Also due to the small viscous effects, especially for inviscid fluids, the slip of tangential velocity at the interface is allowed and only the continuity at normal direction will be considered and is expressed as

$$u_{l,n} = u_{k,n}. \tag{3.43}$$

Since the Lagrangian acceleration of the continuous phase can be expressed as $\boldsymbol{g} - \nabla p_\alpha/\rho_\alpha + v_\alpha \nabla^2 \boldsymbol{u}_\alpha$ which should be continuous as well and the viscous term at the interface is averaged as $(v_l \nabla^2 \boldsymbol{u}_l + v_k \nabla^2 \boldsymbol{u}_k)/2$ as discussed in section 3.1, the normal velocity continuity can be transferred to be

$$\frac{1}{\rho_l}\frac{\partial p_l}{\partial n} = \frac{1}{\rho_k}\frac{\partial p_k}{\partial n}. \tag{3.44}$$

As the pressure is continuous at the interface, the tangential derivative of the pressure at the interface is continuous as well and is expressed as

$$\frac{\partial p_l}{\partial \tau} = \frac{\partial p_k}{\partial \tau}. \tag{3.45}$$

Combining Eq. (3.42) and (3.44), the jump of $\frac{\nabla p}{\rho}$ at the interface is

$$\begin{aligned} &\left(\frac{\nabla p}{\rho}\right)_k - \left(\frac{\nabla p}{\rho}\right)_l \\ &\quad = \left(\frac{\partial p}{\rho \partial n}\right)_k - \left(\frac{\partial p}{\rho \partial n}\right)_l + \left(\frac{\partial p}{\rho \partial \tau}\right)_k - \left(\frac{\partial p}{\rho \partial \tau}\right)_l \\ &\quad = \frac{\partial p}{\partial \tau}\left(\frac{1}{\rho_k} - \frac{1}{\rho_l}\right). \end{aligned} \tag{3.46}$$

The following derivation and discretization will be based on Eq. (3.42) and (3.46) as the interface conditions for fluids with small viscosity and zero interface tension. In the ISPH modelling for multi-phase flow, Shao (2012) only applied the stress continuity without the

consideration of the velocity at the interface. Hu and Adam (2006, 2007) developed an ISPH formulation based on the condition of continuity pressure and $\nabla p/\rho$. However, based on the derivations of Eqs. (3.42) to Eq. (3.46), jumps of $\nabla p/\rho$ at the interface cannot be ignored for flows with high density ratio or significant pressure variation in tangential direction. Another difference in dealing with interface conditions is that, Hu and Adam (2006, 2007) used the concept of inter-particle-averaged directional derivative when they discredited the second order derivatives involved in the Poisson's Equation for pressure. When doing so, they assumed that the middle point between each pair of particles is on the interface. As the gradient and the second order derivatives are determined averagely in their formulation by many pairs of particles that influence the point concerned, the interface in their formulation is actually a layer with a thickness in the order of the radius of the support domain (usually more than double distance of two particles). This is the first attempt to extend the MLPG_R method to dealing with the multiphase flow. A new approach which is not only suitable for the MLPG_R method but also allows applying the conditions described by Eq. (3.42) and (3.46) on a sharp interface is proposed. For this purpose, the pressure near the interface denoted by $\boldsymbol{r}_0$ is firstly expanded into a Taylor series separately in each phase:

$$\frac{1}{\rho_l}\big(p_l(\boldsymbol{r}) - p_l(\boldsymbol{r}_0)\big) \approx \frac{1}{\rho_l}(\nabla p_l)_{\boldsymbol{r}_0} \cdot (\boldsymbol{r} - \boldsymbol{r}_0), \tag{3.47}$$

$$\frac{1}{\rho_k}\big(p_k(\boldsymbol{r}) - p_k(\boldsymbol{r}_0)\big) \approx \frac{1}{\rho_k}(\nabla p_k)_{\boldsymbol{r}_0} \cdot (\boldsymbol{r} - \boldsymbol{r}_0). \tag{3.48}$$

Utilizing Eq. (3.42) and (3.46) and projecting Eq. (3.47) and Eq. (3.48) on normal and tangential directions lead to

$$\frac{1}{\rho_l}\big(p_l(\boldsymbol{r}) - p(\boldsymbol{r}_0)\big) \approx \left(\frac{\partial p}{\rho \partial n}\right)_{\boldsymbol{r}_0}(r_{\boldsymbol{n}} - r_{0n}) + \frac{1}{\rho_l}\left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}(r_{\boldsymbol{\tau}} - r_{0\tau}), \tag{3.49}$$

$$\begin{aligned}\frac{1}{\rho_k}\big(p_k(\boldsymbol{r}) - p(\boldsymbol{r}_0)\big) \approx{}& \left(\frac{\partial p}{\rho \partial n}\right)_{\boldsymbol{r}_0}(r_{\boldsymbol{n}} - r_{0n}) + \frac{1}{\rho_l}\left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}(r_{\boldsymbol{\tau}} - r_{0\tau}) \\ &+ \left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}\left(\frac{1}{\rho_k} - \frac{1}{\rho_l}\right)(r_{\boldsymbol{\tau}} - r_{0\tau}),\end{aligned} \tag{3.50}$$

where $p(\boldsymbol{r}_0)$, $\left(\frac{\partial p}{\rho \partial n}\right)_{\boldsymbol{r}_0}$ and $\left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}$ denote the continuous quantities on the interface.

Secondly, discretizing Eq. (3.49) and Eq. (3.50) in the support domain within each phase yields

$$\frac{1}{\rho_l}\sum_{j=1}^{n}\left(p_l(\boldsymbol{r}_j)-p(\boldsymbol{r}_0)\right)\emptyset_l(\boldsymbol{r}_{j0})$$
$$=\left(\frac{\partial p}{\rho\partial n}\right)_{\boldsymbol{r}_0}\sum_{j=1}^{n}(r_{\boldsymbol{n}}-r_{0n})\emptyset_l(\boldsymbol{r}_{j0})+\frac{1}{\rho_l}\left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}\sum_{j=1}^{n}(r_{\boldsymbol{\tau}}-r_{0\tau})\emptyset_l(\boldsymbol{r}_{j0}), \tag{3.51}$$

$$\frac{1}{\rho_k}\sum_{q=1}^{m}\left(p_k(\boldsymbol{r}_q)-p(\boldsymbol{r}_0)\right)\emptyset_k(\boldsymbol{r}_{q0})$$
$$=\left(\frac{\partial p}{\rho\partial n}\right)_{\boldsymbol{r}_0}\sum_{q=1}^{m}(r_{\boldsymbol{n}}-r_{0n})\emptyset_l(\boldsymbol{r}_{j0})+\frac{1}{\rho_l}\left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}\sum_{q=1}^{m}(r_{\boldsymbol{\tau}}-r_{0\tau})\emptyset_k(\boldsymbol{r}_{q0}) \tag{3.52}$$
$$+\left(\frac{\partial p_k}{\partial \tau}\right)_{\boldsymbol{r}_0}\left(\frac{1}{\rho_k}-\frac{1}{\rho_l}\right)\sum_{q=1}^{m}(r_{\boldsymbol{\tau}}-r_{0\tau})\emptyset_k(\boldsymbol{r}_{q0}),$$

where the shape function $\emptyset(\boldsymbol{r})$ is obtained by the moving least square (MLS) algorithm in a support domain containing both phases, n and m are the numbers of particles within the support domain in phase l and phase k respectively.

Thirdly, adding up Eq. (3.51) and (3.52) yields

$$\frac{1}{\rho_l}\sum_{j=1}^{n}p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0})+\frac{1}{\rho_k}\sum_{q=1}^{m}p_k(\boldsymbol{r}_q)\emptyset_k(\boldsymbol{r}_{q0})$$
$$-\left(\frac{1}{\rho_l}p(\boldsymbol{r}_0)\sum_{j=1}^{n}\emptyset_l(\boldsymbol{r}_{j0})+\frac{1}{\rho_k}p(\boldsymbol{r}_0)\sum_{q=1}^{m}\emptyset_k(\boldsymbol{r}_{q0})\right)$$

$$=\left(\frac{\partial p}{\rho\partial n}\right)_{\boldsymbol{r}_0}\left[\sum_{j=1}^{n}(r_{\boldsymbol{n}}-r_{0n})\emptyset_l(\boldsymbol{r}_{j0})+\sum_{q=1}^{m}(r_{\boldsymbol{n}}-r_{0n})\emptyset_k(\boldsymbol{r}_{j0})\right] \tag{3.53}$$
$$+\frac{1}{\rho_l}\left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}\left[\sum_{j=1}^{n}(r_{\boldsymbol{\tau}}-r_{0\tau})\emptyset_l(\boldsymbol{r}_{j0})+\sum_{q=1}^{m}(r_{\boldsymbol{\tau}}-r_{0\tau})\emptyset_k(\boldsymbol{r}_{q0})\right]$$
$$+\left(\frac{\partial p}{\partial \tau}\right)_{\boldsymbol{r}_0}\left(\frac{1}{\rho_k}-\frac{1}{\rho_l}\right)\sum_{q=1}^{m}(r_{\boldsymbol{\tau}}-r_{0\tau})\emptyset_k(\boldsymbol{r}_{q0}).$$

The first two terms on the RHS of Eq. (3.53) could be cancelled out and become zero when sufficient particles are applied. Even though the disretization of the last term is taken only within one phase (on partial support domain), $\sum_{q=1}^{m}(r_{\boldsymbol{\tau}}-r_{0\tau})\emptyset_k(\boldsymbol{r}_{q0})$ also becomes zero when sufficient particles are used due to the symmetricity of $(r_{\boldsymbol{\tau}}-r_{0\tau})$ to

the centre particle. However, to consider general particle distributions, RHS of Eq. (3.53) will be retained and is represented by F_r. Rearranging the Eq. (3.53) yields

$$P(\boldsymbol{r}_0) - \frac{\rho_k \sum_{j=1}^{n} p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0}) + \rho_l \sum_{q=1}^{m} p_k(\boldsymbol{r}_q)\emptyset_k(\boldsymbol{r}_{q0}) - F_r}{\rho_k \sum_{j=1}^{n} \emptyset_l(\boldsymbol{r}_{j0}) + \rho_l \sum_{q=1}^{m} \emptyset_k(\boldsymbol{r}_{q0})} = 0, \tag{3.54}$$

where the pressure gradient will be estimated with the pressure of the last time step. Eq. (3.54) provides a simple and explicit pressure expression on the interface which is obtained by imposing the two conditions in Eq. (3.42) and Eq.(3.46). If we would employ the same formulation as in Shao (2012) ensuring only pressure continuity at the interface and obtaining the interface pressure by interpolating from the neighbouring particles, the pressure should be satisfied as

$$P_{DISPH}(\boldsymbol{r}_0) - \frac{\sum_{j=1}^{n} p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0}) + \sum_{q=1}^{m} p_k(\boldsymbol{r}_q)\emptyset_k(\boldsymbol{r}_{q0})}{\sum_{j=1}^{n} \emptyset_l(\boldsymbol{r}_{j0}) + \sum_{q=1}^{m} \emptyset_k(\boldsymbol{r}_{q0})} = 0, \tag{3.55}$$

where the subscript $DISPH$ represents the decoupled ISPH (Shao 2012).

It can be seen that Eq. (3.54) can be simplified to Eq. (3.55) only if $\rho_k/\rho_l \to 1$, i.e. for the cases of two phases with similar densities and $F_r = 0$ (could be satisfied if the number of particles is sufficient). It maybe deduced that the interface condition of pressure continuity without Eq. (3.46) is suitable only for low density ratios and the extension to high ratios needs an additional condition of $\nabla p/\rho$. The major difference between Eq. (3.54) and the approach in Hu and Adams (2006; 2007) lies that (a) $\nabla p/\rho$ is jumped at the interface rather than continuous based on the conditions of pressure continuity and normal velocity continuity; (b) Eq. (3.54) is applied on the interface (curve in 2D) rather than in a layer, so potentially the interface between two phase can be very sharp, and (c) no second order derivatives are involved here.

It is noted that if the viscosity is not small and/or the surface tension needs to be considered, the conditions in Eq. (3.42) and (3.46) have to be modified. The approach described here can be extended to deal with flows containing large viscous fluids with significant interface tension by modifying the interface conditions. This will be considered in chapter 7.

3.4 Pressure formulation of isolated particles

As indicated above, it is assumed that a small number of isolated particles have no significant effects on the pressure of continuous phase, and so the pressure at the position occupied by an isolated particle can be estimated by considering that the position is occupied by a particle of the surrounding phase, i.e., it may be given by

$$P\left(\boldsymbol{r}_{isp}\right) - \frac{\sum_{j=1}^{n} p_l\left(\boldsymbol{r}_j\right)\emptyset_l\left(\boldsymbol{r}_{j0}\right) - F_r}{\sum_{j=1}^{n} \emptyset_l\left(\boldsymbol{r}_{j0}\right)} = 0, \tag{3.56}$$

where $\boldsymbol{r}_{isp}$ denotes the position of the isolated particle, l is the other phase surrounding the isolated particle. There are many other ways to replace Eq.(3.56), including the direct application of Eq. (3.16) by temporally assigning the surrounding density to the isolated particle, which has also been tried and gave similar results. As Eq. (3.56) is easy to be implemented and consistent with Eq. (3.54), i.e., Eq. (3.54) becomes Eq. (3.56) by setting $m = 0$, the Eq. (3.56) is kept for use. Due to its consistency with Eq. (3.54), the equation for the isolated particle is actually treated as a special case for the interface particle but affected effectively by one phase.

Chapter 4 Technique of Identifying Interface Particles

In numerical simulations of two-phase flows, the description of interface evolution is essential to couple the two phases in solving equations describing the transport of momentum, mass and energy. In multiphase MLPG_R method, to explicitly implement the interface conditions as expressed by Eq. (3.42) and Eq. (3.46) or further development by considering interface tension and significant viscous effects, interface particles have to be first identified. There are various methods for interface description which can be broadly divided into two groups. One is based on mesh methods usually representing the interface by an indicator function resulting in the thickness either covering several grid cells or sharpened to zero. The other group is based on meshless methods, in which particles are moved in a Lagrangian manner and the surface or interface boundaries are represented directly.

In mesh based methods, the volume of fluid (VOF) method captures the interface by the indicator function formulated with volume fraction of either phase in each mesh. To represent a sharp interface within the mesh, VOF is often complemented by reconstruction techniques such as piece wise constant approach (Hirt & Nichols 1981; Yokoi 2007), Piecewise linear approach (Aulisa et al. 2007; Rudman 1997) in which the interface in a mesh is assumed to be a line in 2D or a plain in 3D, piecewise quadratic (Diwakar et al. 2009; Renardy & Renardy 2002)or piecewise cubic spline (Ginzburg & Wittum 2001; López et al. 2004).To simplify the computational treatment, the interface reconstruction is subtracted representing an interface covering several meshes (Rudman

1997; Ubbink & Issa 1999; Xiao et al. 2005). And later on the artificial compression terms were introduced (Weller 2008; Hoang et al. 2013) to sharpen the interface and counteract the numerical diffusion. Different from artificial geometric representation in VOF when reconstructing the interface, the indicator in level-set method (Osher & Sethian 1988) is a smoothed signed distance function from the interface and zero level-set gives the location of the interface. To remedy its mass conservation problem, reinitializations for the signed distance function (Sussman & Fatemi 1999; Min 2010) and mass corrections (Ausas et al. 2011; Zhang et al. 2010) were widely adopted. Instead of using Eulerian mesh, the arbitrary Lagrangian-Eulerian mesh system was adopted in which the unstructured meshes are moved to keep the grid nodes at the interface (Muzaferija & Peri´c 1997; Tukovic & Jasak 2012).

As for meshless particle methods, interface can be inherently tracked by moving particles which are initially allocated to one phase and maintained during the simulation. Without connections between particles, the treatment of large deformation, fragmentation and coalescence of the interface or surface becomes much easier. For methods that solve two phases in one set of equations, in both SPH (Hu & Adams 2007; Chen et al. 2015)and MPS (Shakibaeinia & Jin 2012a; Khayyer & Gotoh 2013), interface conditions are implicitly implemented and thus the interface does not need to be explicitly identified. But for explicit boundary condition implementation, two situations can be observed. One situation is that the flow is not violent or large deformations do not occur, under which particles on free surface in single phase flows or on the interface in multiphase flows are maintained during the simulation (Ma 2005b). The other situation comes with flows being violent or with large deformations and therefore the inner and boundary particles are changeable (Lee et al. 2008; Ma & Zhou 2009; Shao 2012). Thus, surface or interface identification is not necessary for the former situation but for the latter it has significant impact on boundary condition implementation and consequently on final results.

In this chapter a new interface particle identification technique is proposed based on absolute density gradient for violent flows even with isolated particles penetrating into the other phase. In section 4.1 the existing methods for surface and interface particle identification will be briefly reviewed. Then the formulations of the new identification technique and parameter tests are given in section 4.2. In section 4.3, comparisons between the new technique and the existing ones are first carried out and then followed by the validations of the new technique on different density ratios, increasing randomness of particle distribution, various sizes of the support domain in calculating the density gradient and flow conditions.

4.1 Existing methods for identifying interface nodes

Many applications of meshless methods (e.g. incompressible SPH (ISPH), MPS, MLPG) in single phase wave simulations rely strongly on free surface particle identification based on which the dynamic surface boundary condition can be implemented. A number of approaches have been proposed and adopted in various meshless methods, such as these using the particle number density method (abbreviated to PND) in MPS (Koshizuka et al. 1998; Gotoh & Sakai 2006; Xu & Jin 2014) and fluid density in ISPH (Shao 2009). Both techniques are developed mainly for single phase problems, which lead to lower particle number density (MPS) or lower density (SPH) on the free surface, satisfying

$$n_i < \theta n^0, \tag{4.1}$$

$$\rho_i < \kappa \rho^0, \tag{4.2}$$

respectively, where $n_i = \sum_j w(|\boldsymbol{r}_j - \boldsymbol{r}_i|)$ is the particle number density for the target article i with neighbouring ones j, and n^0 denotes the inner particle number density which is initially defined. Similar to particle number density, $\rho_i = \sum_j m_j w(|\boldsymbol{r}_j - \boldsymbol{r}_i|)$ is the estimated density and ρ^0 denotes the density of the inner particle. θ and κ are coefficients to evaluate the decrease in n_i and ρ_i for the surface particles. As pointed out by (Koshizuka & Oka 1996) the value of θ could be in the range of 0.8 to 0.99 and was selected as 0.97 in their case study of dam breaking and in (Xu & Jin 2014) for weir flows. n_i and κ were also chosen as 0.95 for breaking waves on slopes (Koshizuka et al. 1998) and 0.99 for water entry of a free-fall object (Shao 2009). However, this simple approach has been found often to identify incorrectly the interface particles, especially when the particles are unevenly distributed.

As a supplementary to PND, a mixed particle number density and auxiliary function method (MPAM) was proposed (Ma & Zhou 2009) by additionally counting the existing surface particles and the occupied quarters and crossed rectangles in the support domain (Eq. (22)-(24) and Fig.3 in (Ma & Zhou 2009)). Following Shao (2009) and Ma & Zhou (2009), the method was further developed by Zheng et al. (2014) by defining two quarter-dividing systems and counting the occupied ones (Eq. (28)-(30) and Fig.4 in (Zheng et al. 2014)) in which the value of κ was chosen as 0.9. As confirmed by many calculations in the cited papers, the accuracy of identifying interface particles is significantly improved with the auxiliary functions. However apart from the complexity in sorting neighbouring particles into quarters and other specific regions, the dependence on the identification of

the previous time step could lead to error accumulation once misidentification occurs in previous steps.

Lee et al. (2008) andRafiee et al. (2012) employed a tracking method by estimating the divergence of position vector, $\boldsymbol{r}$, with the expression of

$$(\nabla \cdot \boldsymbol{r})_i \approx \sum_j \Delta V_j (\boldsymbol{r}_j - \boldsymbol{r}_i) \cdot \nabla w(|\boldsymbol{r}_j - \boldsymbol{r}_i|) \tag{4.3}$$

where ΔV_j is the volume of particle associated with particle j and w is a weight function. The value of the numerical divergence from Eq. (4.3) is close to 2 for inner particles in two dimensional simulations and decreases on the free surface particles for the reason that the number of neighbouring particles is smaller and the support of the kernel is truncated. Thus the criterion identifying the free surface particles is set as

$$(\nabla \cdot \boldsymbol{r})_i < 1.5. \tag{4.4}$$

It was indicated that not all the surface particles were identified (Lee et al. 2008) and some inner particles are over identified (Lind et al. 2012) although the misjudgements appeared to be insignificant in the cases they have considered by using the method. Because the theoretical value of the position vector divergence should be 2 for both inner and surface particles, its numerical value should be close to 2 even near or on the interface if more accurate and consistent approximation, such as SFDI gradient scheme (Ma 2008) is employed to estimate the divergence. From this point of view, the method is questionable for its general use and so will not be considered further in this paper.

Above surface particle identification methods can be applied in a straightforward way to interface identification in two-phase flows if the density difference between the two phases are large (Shao et al 2012) or by neglecting the additional phase and picking out the ‘surface’ of the other phase. To improve the accuracy and reduce the complexity of the interface identification technique, a new approach based on the absolute density gradient is proposed in this paper. This method is based on the fact that the density gradient is infinite on the interface but zero away from it for incompressible fluids. In two-phase flows, it often happens that one phase penetrates into the other especially in violent flow situations. In that case, the penetrated parts are usually represented by an isolated particle for which special treatment is necessary (Gotoh & Sakai 2006). Gotoh & Sakai (2006) also provided an identification method using a combination of particle number density and the number of neighbouring particles of the other phase. In

comparison the newly proposed technique is able to identify such isolated particles in a more consistent way.

4.2 Absolute density gradients (ADG) based technique

For immiscible two-phase flows with a sharp density jump at the interface, the density gradients are 0 for inner particles and become theoretically infinite at the interface if both phases are incompressible. For weak compressible phases, the density gradients are close to 0 for inner particles and are also theoretically infinite at the interface. To utilize this fact for interface identification, the absolute value of density gradient is numerically computed as below

$$\beta = |\rho_{,x}| + |\rho_{,y}|, \tag{4.5}$$

where

$$|\rho_{,d}| = \frac{2}{\sum_j w(\boldsymbol{r}_j - \boldsymbol{r}_i)} \sum_j |\rho_i - \rho_j| \frac{|r_{j,d} - r_{i,d}|}{\Delta r_{ij}{}^2} w(\boldsymbol{r}_j - \boldsymbol{r}_i), \quad d = x, y \tag{4.6}$$

and the weight function $w(\boldsymbol{r}_j - \boldsymbol{r}_i)$ is selected as

$$w(\boldsymbol{r}_j - \boldsymbol{r}_i) = \begin{cases} 1 - 6\bar{r}^2 + 8\bar{r}^3 - 3\bar{r}^4 & \bar{r} = \dfrac{r_{ij}}{R_e} = \dfrac{|\boldsymbol{r}_j - \boldsymbol{r}_i|}{R_e} \leq 1 \\ 0 & \bar{r} = \dfrac{r_{ij}}{R_e} = \dfrac{|\boldsymbol{r}_j - \boldsymbol{r}_i|}{R_e} > 1 \end{cases}, \tag{4.7}$$

where $\boldsymbol{r}_j$ and $\boldsymbol{r}_i$ are the position vectors of neighbour and target particles, $r_{j,x}$ and $r_{j,y}$ are the component of $\boldsymbol{r}_i$ in the x- and y- direction, R_e is the radius of the support domain proportional to Δl with Δl being the initial average particle distance and Δr_{ij} is defined as $\Delta r_{ij} = max(0.8\Delta l, |\boldsymbol{r}_j - \boldsymbol{r}_i|)$.

The numerical value of β is very close 0 for inner particles and it rapidly increases when approaching to the interface or isolated particles. Although the gradient is theoretically infinite at the interface or isolated particles, it is just a large value in numerical computation.

To obtain a non-dimensional scale of the absolute density gradient β_0 is introduced with the expression of

$$\beta_0 = |\rho_1 - \rho_2|/\Delta l \tag{4.8}$$

where ρ_1 and ρ_2 are densities of two phases. The interface and isolated particles will be identified by the ratio of β/β_0 rather than β. To shed some light on the order of the ratio, we may consider the special cases: uniformly distributed particles with some of them being on horizontal (or vertical) interface, diagonal interface or isolated. If only the nearest four neighbours are considered (i.e., the support domain is in the range of $1.0\Delta l \Delta l$ and $\sqrt{2}\Delta l$), the value of β/β_0 should be 0.5 for horizontal or vertical interfaces, 1.0 for diagonal interfaces and 2.0 for isolated particles by using Eq. (4.5) and (4.6). It is noted here that for any fixed distribution of particles with the weight function given by Eq. (4.7) and a fixed support domain size, the value of β/β_0 remains independent on the value of Δl. It is also noted that the value of β/β_0 will not be affected by the density ratio of the two phases. In other words, the value of β/β_0 is independent of the total number of particles for a given domain and density ratio. Nevertheless, in general situations with the particular weight function used and the chosen radius of the support domain, the value of β/β_0 will vary in a range with the change of particle distribution. Therefore most suitable values need to be determined by considering a range of flow conditions of the two phases. The following two conditions (I) and (II) are then checked:

(I) $\gamma < \beta/\beta_0 < \alpha,$ (4.9a)

(II) $\beta/\beta_0 \geq \alpha.$ (4.9b)

If Condition (I) is satisfied, the particles are identified to be interface particles. If Condition (II) is met, the particles are identified as isolated particles. If none of the conditions is satisfied, they are classified as inner particles. For general cases, particles move and become irregularly distributed. The values of γ and α will be discussed below using numerical tests. This approach based on Eqs. (4.5) to (4.9a) is shortened as ADG (absolute density gradient) method hereafter for convenience.

To determine the value of γ and α, a number of tests are carried out for a specified particle configuration as shown in Figure 4.1. In this figure, two typical regions are enlarged. Particles are clustered in region 1 and relatively coarser in region 2. As shown in Figure 4.1(a), an under-specified value of γ (e.g. 0.1), leads to misidentifying inner particles close to the interface in the clustered region (some inner particles are identified as interface particles, i.e., over-identified). However, an over-specified value of γ (e.g. 0.5), shown in Figure 4.1(b), leads to missing the particles on the interface in the region where the distribution of particles is relatively coarser (i.e., under-identified). Based on

this, we can deduce that there might be a range of values for γ which may lead to correctly identifying the interface particles.We have tested the value of γ in the range of 0.2 to 0.4 for the same configuration and found that the interface particles are correctly picked out and insensitive to the specific value in the range. The results of $\gamma = 0.2$ and $\gamma = 0.4$ are illustrated in Figure 4.2. Hereafter, the value of γ is selected to be 0.3, unless mentioned otherwise.

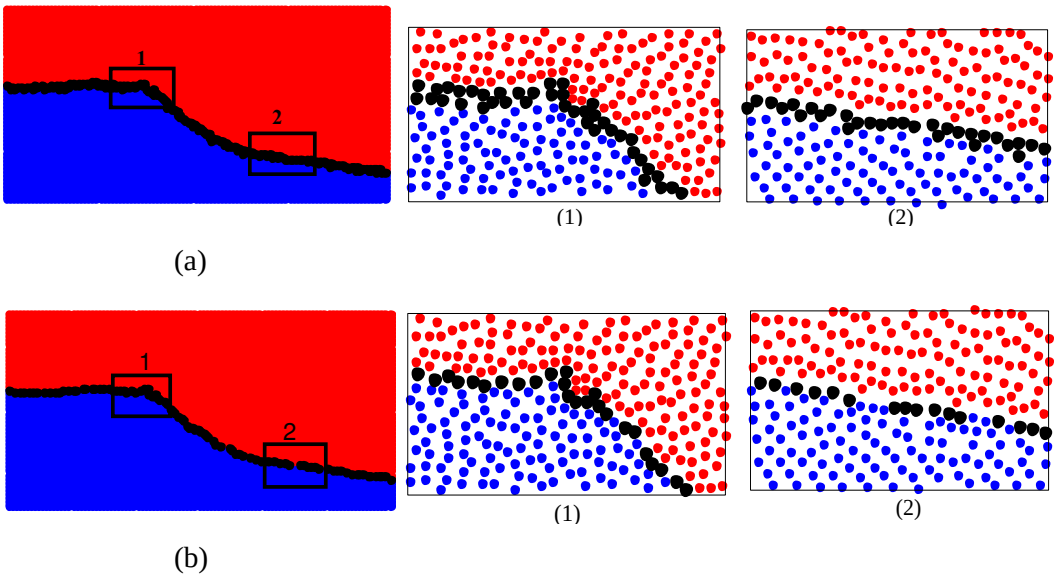


Figure 4.1: Examples of interface identifications when applying improper value of γ. (a) shows inner particles close to the interface are over-identified with $\gamma = 0.1$ and (b) shows some interface particles are under-identified with $\gamma = 0.5$. (density ratio: 1:1000)

Another particle configuration associated with the violent sloshing of two phases is used to illustrate the effects of value for α in Condition (II). Figure 4.3(a) and (b) show that the isolated particles penetrating into the other phase are identified by using the value of α to be 1.4 and 1.6, respectively. It can be seen that the isolated particles of both phases are correctly identified by using either value. This demonstrates that the results are not sensitive to the value of α when it is within the range of 1.4 to 1.6. Thus α=1.5 will be used for isolated particle identification.

According to above numerical tests, the values of γ and α are selected to be 0.3 and 1.5 for the conditions of (I) and (II) to identify the interface particles and isolated particles respectively.

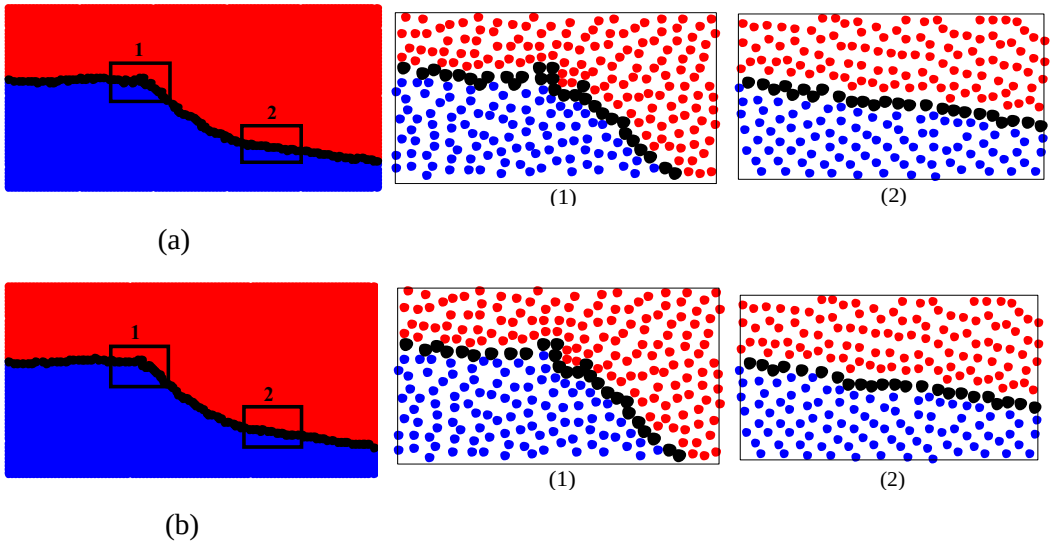


Figure 4.2: Examples of interface identifications when applying proper values of γ. (a) shows the interface identified by $\gamma = 0.2$ and (b) is obtained by $\gamma = 0.4$. (density ratio: 1:1000)

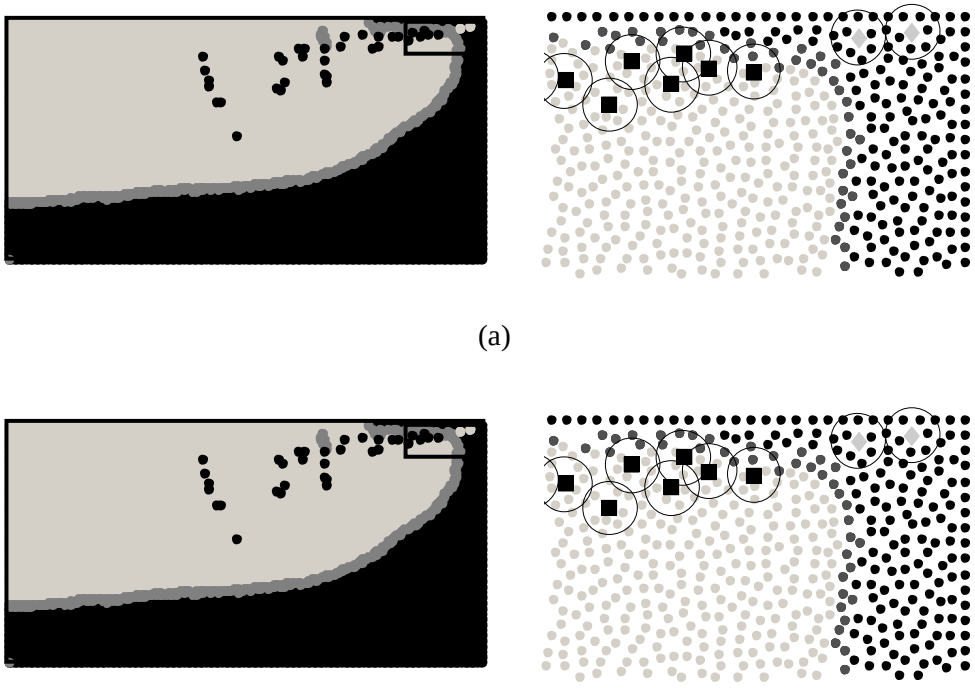


Figure 4.3: Isolated particles marked by squares for heavier fluid particles and diamonds for lighter fluid particles (density ratio: 1:1000) are identified with α being 1.4 (a) and 1.6 (b). The heavier phase is marked in black and the lighter is in light grey. In the enlarged figures on the right, support domains for each isolated particles are marked with a circle.

4.3 Validations of the ADG method

4.3.1 Identification with different random distributions of particles

The newly proposed interface identification technique will be further validated on random distributed particles. Comparison will be made with the results of other two methods. One is the PND method used by ISPH simulation and the other is the MPAM suggested by Ma & Zhou (2009) as mentioned in section 4.2. The tests are first carried out on the specified configuration shown in Figure 4.4 in which the unit patch is divided into two portions by a sinusoidal curve of $y = 0.1sin(2\pi x) + 0.5$. Particles are randomly distributed by using the 'haltonset' function available in MATLAB. Below the sinusoidal curve the particles represent the fluid phase 1 (blue dots) and the rest is filled by fluid phase 2 (red dots).

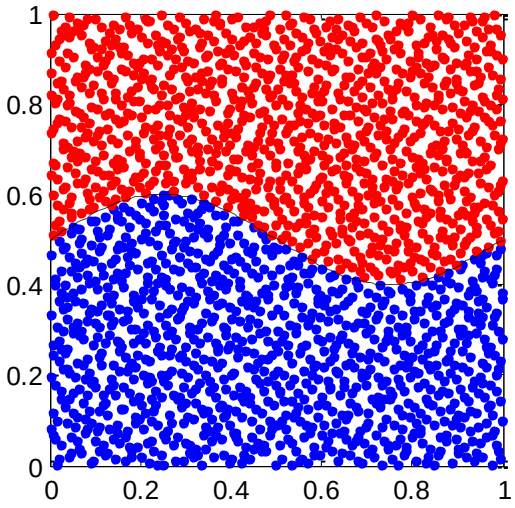


Figure 4.4: Sketch of interface testing case. 2000 random distributed particles in a unit patch divided into two portions by $y = 0.1sin(2\pi x) + 0.5$ shown by dashed line. Blue and red dots represent the fluid phase 1 and fluid phase 2, respectively.

For the methods of the PND and MPAM, interface particles are identified as surface particles of the phase 1 by ignoring the presence of the phase 2 as they are developed for the single phase problem. As shown in Figure 4.5 green dots are interface particles identified by the PND method with $\theta = 0.85$ and $\theta = 0.97$ in Figure 4.5(a) and (b), which are in the range of θ proposed by (Koshizuka & Oka 1996). One can observe that the interface particles are highly over-identified as many inner particles are marked as interface ones although it is moderately relieved by decreasing the coefficient. Such over-identification was also observed by (Zheng et al. 2014) in their dam breaking simulation.

There are also several particles located on the interface but not identified by using either value of θ. Other θ values are also tested and the results are shown in Figure 4.6. One can find that decreasing the value helps alleviate over-identification of inner particles but more interface particles are also missed. The reasons for such problems are that the accuracy of the PND method strongly depends on the randomness of the particle distribution which was pointed out by (Ma & Zhou 2009). For inner particles, coarse distribution leads to low particle number density even with complete support domain and so over-identification happens. On the contrary, particle clustering on the interface leads to high particle number density even though the support domain of its own phase is incomplete. For the results obtained using the MPAM shown in Figure 4.5(c), the accuracy is significantly improved. But over identification still happens on inner particles close to the interface. Since this identification technique is based on the interface particles identified at the previous time step, error accumulation may occur after long time simulation. Figure 4.5(d) shows the results obtained by the method proposed in this paper. It can be seen that almost all the interface particles are correctly identified by employing a pre-tested coefficient of $\gamma = 0.3$ and a clear sharp interface is obtained. No inner particles are wrongly identified by the method. As indicated above, the new technique does not depend on the density ratio. To confirm this, the tests are also carried out on different density ratios of 0.1 and 0.01 in addition to the ratio of 0.9 in Figure 4.5(d) by using the ADG method, and the results for the density ratios of 0.1 and 0.01 are shown in Figure 4.7. This figure and Figure 4.5(d) show that the accurate identification is achieved in all the cases, demonstrating that the ADG method results are independent of the density ratios.

In modelling violent multiphase flow, the particle distribution can become very random even they are uniformly distributed initially. It is important that the behaviours of interface identification methods should be examined for different levels of randomness of particle distributions. For this purpose, we use a configuration similar to that in Figure 4.4 with particle number of 2500; however the particles are distributed in a way that they are first uniformly located and then deviated by $\Delta \boldsymbol{r} = k\left[\left(R_{n_x} - 0.5\right)\boldsymbol{e}_x + \left(R_{n_y} - 0.5\right)\boldsymbol{e}_y\right]\Delta l$, i.e., the position of particle is given by $\boldsymbol{r} = \boldsymbol{r}_0 + \Delta \boldsymbol{r}$ where each of $R_{n_x}$and R_{n_y} is a group of random numbers ranging from 0 to 1.0 generated separately in x- and y-direction respectively; k is the randomness and Δl is the particle distance in uniform distribution; $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ are the unit vector in x- and y-direction, respectively; k leads to zero deviation and uniform distribution.

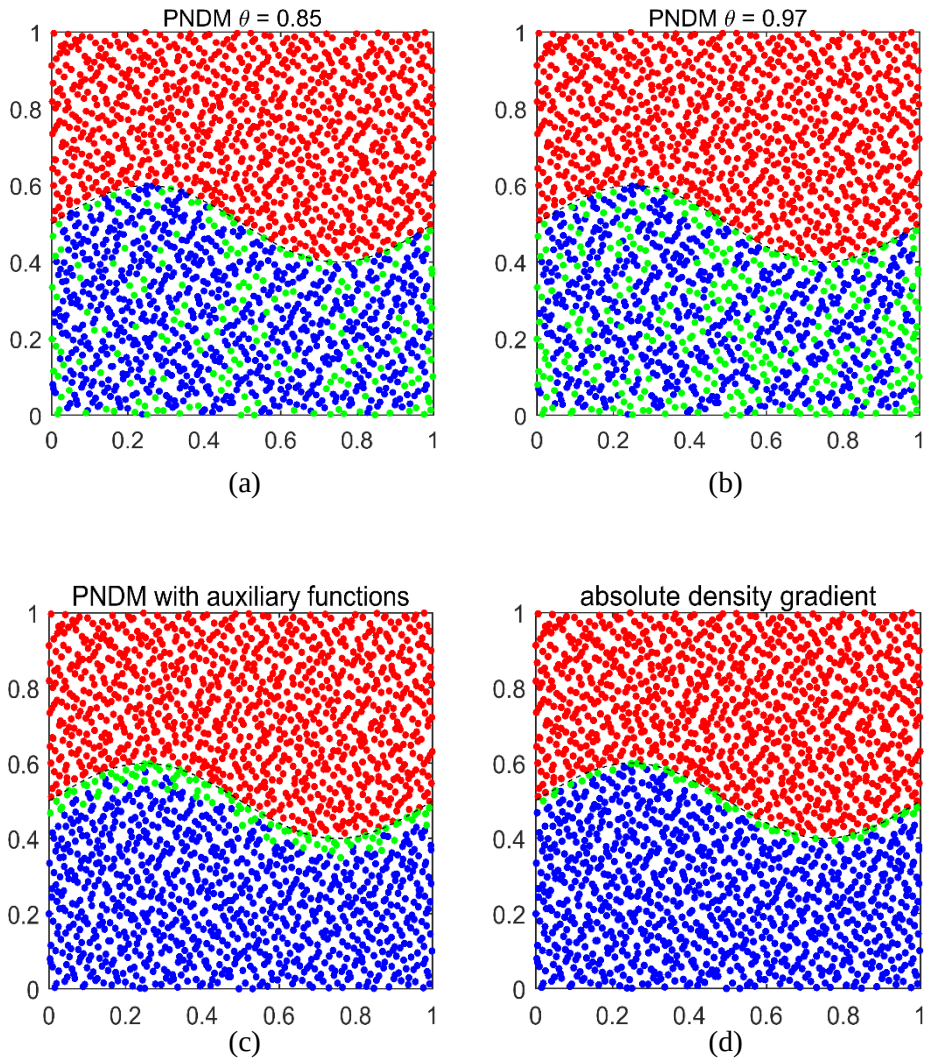


Figure 4.5: Interface particles (green nodes) identification by PND with $\theta = 0.85$ (a), PND with $\theta = 0.97$(b), PND with auxiliary functions (c) and ADG (d).Blue and red dots represent the fluid phase 1 and fluid phase 2 respectively. (density ratio=0.9).

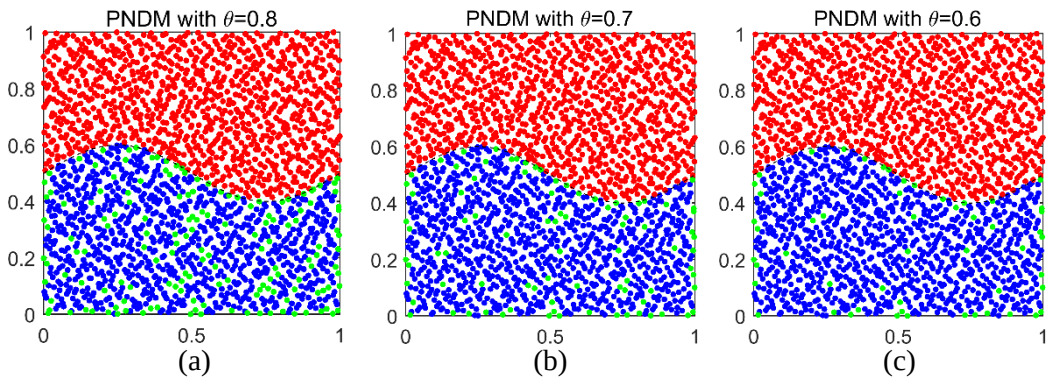


Figure 4.6: Interface particles (green nodes) identification by PND with $\theta = 0.8$ (a), $\theta = 0.7$(b) and $\theta = 0.6$ (c). Blue and red dots represent the fluid phase 1 and fluid phase 2, respectively.

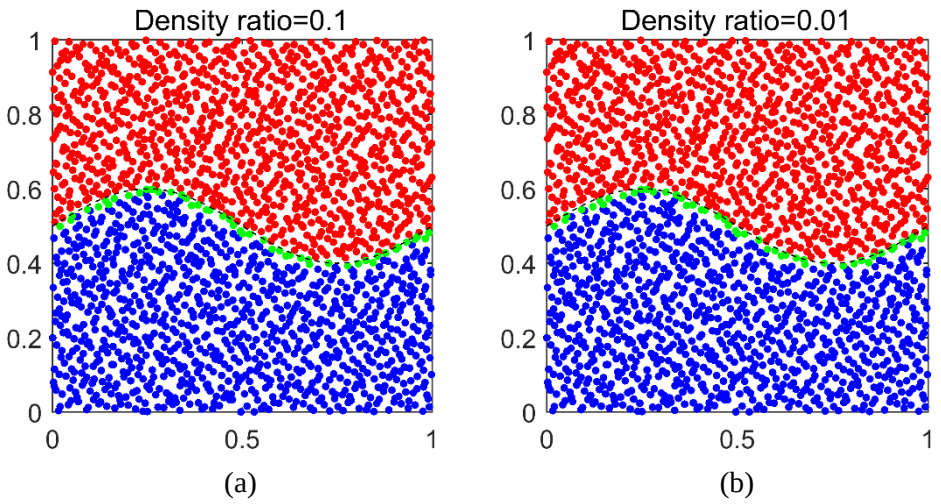


Figure 4.7: Interface particles (green nodes) identification by ADG technique on density ratios of 0.1 with the densities of blue and red dots of 10 and 1 respectively in (a) and 0.01 with the densities of blue and red dots of 100 and 1 respectively in (b).

As k increases, the distribution becomes more disorderly. If the value of k is equal to or large than 1, there would be possibility that the position of a particle can be shifted by more than Δl. The three methods (PND, MPAM and ADG) are employed to identify the interface particles for the cases with different level of randomness. The numbers of particles inaccurately identified for three techniques are listed in Table 4.1 with k increasing from 0 to 1.2. The reference number is the number of particles at or near the specified interface curve within the distance of $0.5\Delta l$ for different randomness of k. One can see that the number of misidentified particles by the PND method is small only when the value of k is small but with the increase of the k value (level of randomness), the number of misidentified particles is large, even much larger than the real number of interface particles. In contrast, the number of misidentified particles for each value of k by the MPAM is significantly reduced but is still considerable. Comparatively, the number of misidentification by the ADG method is very small and does not increase with the increase of randomness. To have a clearer look at their performances, the ratio of inaccurate identification defined as the number of misidentified particles divided by the reference number is demonstrated in Figure 4.8. It shows that, by increasing the randomness, more than 7 times of the specified interface particles are misidentified by the PND when k reaches 1.2 while this ratio decreases to 0.97 for the MPAM. The maximum ratio is only 0.125 arising at $k = 1.0$ and shows less dependency on the value of k by the ADG. With the same configuration, the particle numbers of 900, 1600, 2500, 3600, 4900 and 6400 are also tested using the ADG method with the randomness $k = 0.6$. The results shown in Figure 4.9 illustrate similar performance with the ratio of inaccurate identification ranging from 3.6% to 6.8% as the particle number varies, indicating that the

ADG technique is insensitive to the particle number. Another important thing worth noting is that a small number of the misidentified particles by the ADG are all near the interface while a large number of inner particles far away from the interface are misidentified by the PND as shown in Figure 4.10(a) and (b). The misidentification of particles near the interface will not significantly affect the computation results as the values of physical variables near the interface are very close to those on the interface. However, the misidentification of particles far away from the interface will significantly alter the results of physical variables. More discussions about this point may be found in Ma and Zhou (2009).

Table 4.1: The number of misidentified particles for different particle distribution randomness by using PND, MPAM and ADG methods

Randomness k	Reference No.*	Number of misidentified particles		
		PND	MPAM	ADG
0	37	0	2	0
0.2	42	54	16	0
0.4	40	173	17	0
0.6	37	222	22	1
0.8	36	232	30	3
1.0	32	222	30	2
1.2	33	232	32	1

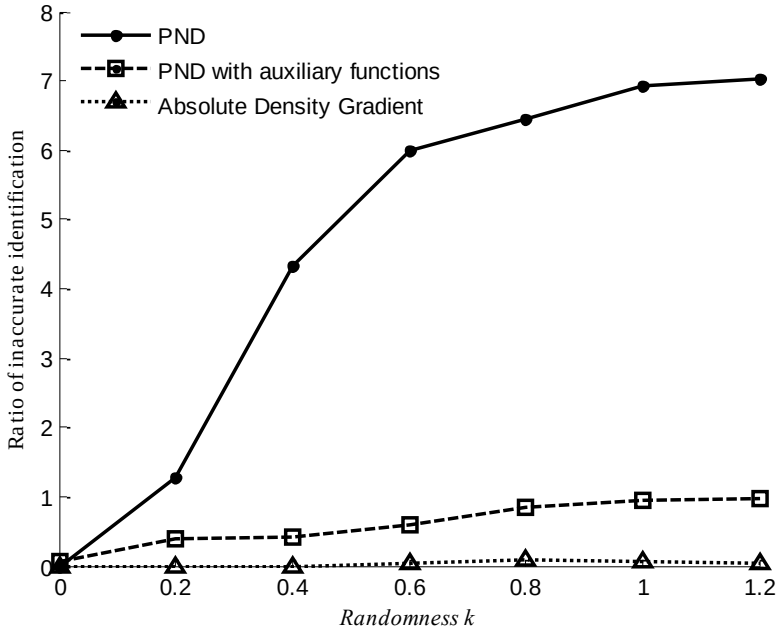


Figure 4.8: The ratios of inaccurate identification (number of misidentified particles/the reference number) of increasing randomness by using three identification techniques.

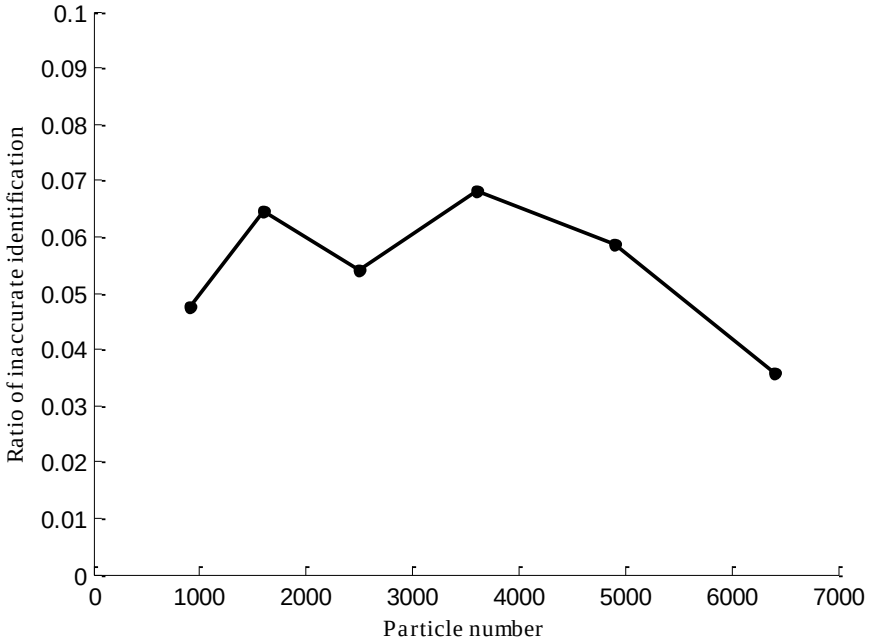


Figure 4.9: The ratios of inaccurate identification (number of misidentified particles/the reference number) of increasing particle number using the method of ADG with randomness of 0.6.

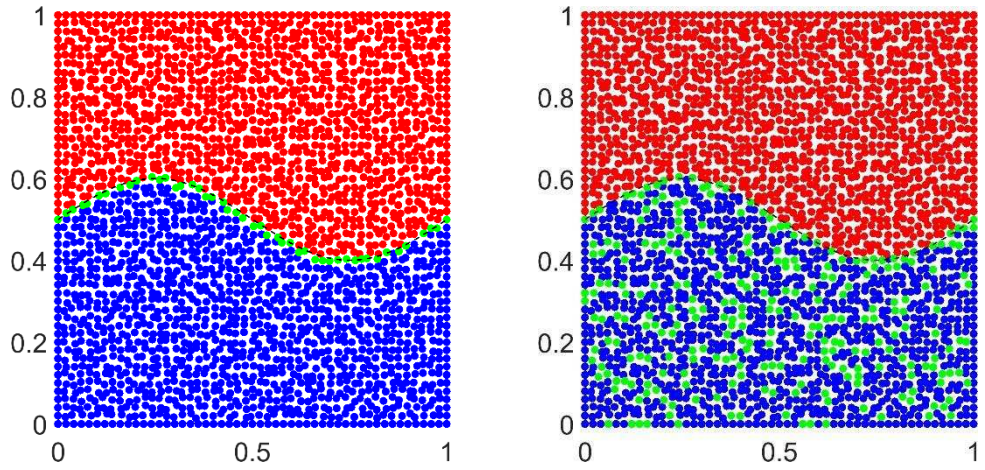


Figure 4.10: Interface particles (green nodes) identification by ADG (a) and PND (b) with particle number of 2500 and randomness of $k = 0.6$. Blue and red dots represent the fluid phase 1 and fluid phase 2 respectively

Tests are also carried out for various support domain sizes R_e of the weight function within the range of $1.4\Delta l$ to $4.0\Delta l$ which is widely adopted for function interpolation, gradient and Laplacian approximations (Ataie-Ashtiani & Farhadi 2006; Gotoh & Fredsøe 2000; Xu & Jin 2014; Zheng et al. 2014). Following the above setup for $k = 0.6$ and 1.0, the numbers of misidentified particles by the ADG are listed in Table 4.2. It shows that a small support domain (e.g. $R_e/\Delta l = 1.4, 1.6$) leads to moderate increment of misidentification due to insufficient neighbouring particles to accurately estimate the density gradient. By increasing $R_e/\Delta l$ (>2.1), the results become insensitive to the variation of the size and even better. Considering the computational time, $R_e/\Delta l = 2.1$ as selected in the above tests is reasonable, but one can choose to use a larger support domain for identifying the interface particles.

Table 4.2: Numbers of misidentified particles by the ADG method with different support domain size R_e for two values of $k = 0.6$ and 1.0

$R_e/\Delta l$	1.4	1.6	1.8	2.1	2.5	3.0	3.5	4.0
$k = 0.6$	7	6	2	1	0	0	0	0
$k = 1.0$	8	6	3	2	1	1	0	0

4.3.2 Identifications for the dam breaking flow

To further validate the newly proposed identification technique on different flow conditions, more complex interface profiles are tested. To focus on the testing of the

identification technique, rather than complete simulation, interface snapshots are picked from a dam breaking simulation in Zheng et al. (2014), based on which particles are randomly distributed by using the 'haltonset' function available in MATLAB and particle densities are assigned based on the interface profiles provided by the simulation in in Zheng et al. (2014). The heavier fluid (phase 1-water) is indicated by blue dots and the lighter fluid (phase 2 – air) is indicated by red dots. Based on such density distribution, interfaces are re-identified by the new technique and their profiles are shown in Figure 4.11 including the changing and irregular shape of the released water with a smooth interface in (a), a backward jet in (b), the jet merging into the bottom water in (c) and a second jet in (d). A trapped air bubble also appears in (c) and (d). The density ratio of the two phases in this test is 0.001. As we can observe from Figure 4.11 interface and isolated particles (black dots) are accurately identified for all the configurations, indicating that the ADG method works well for more complex cases.

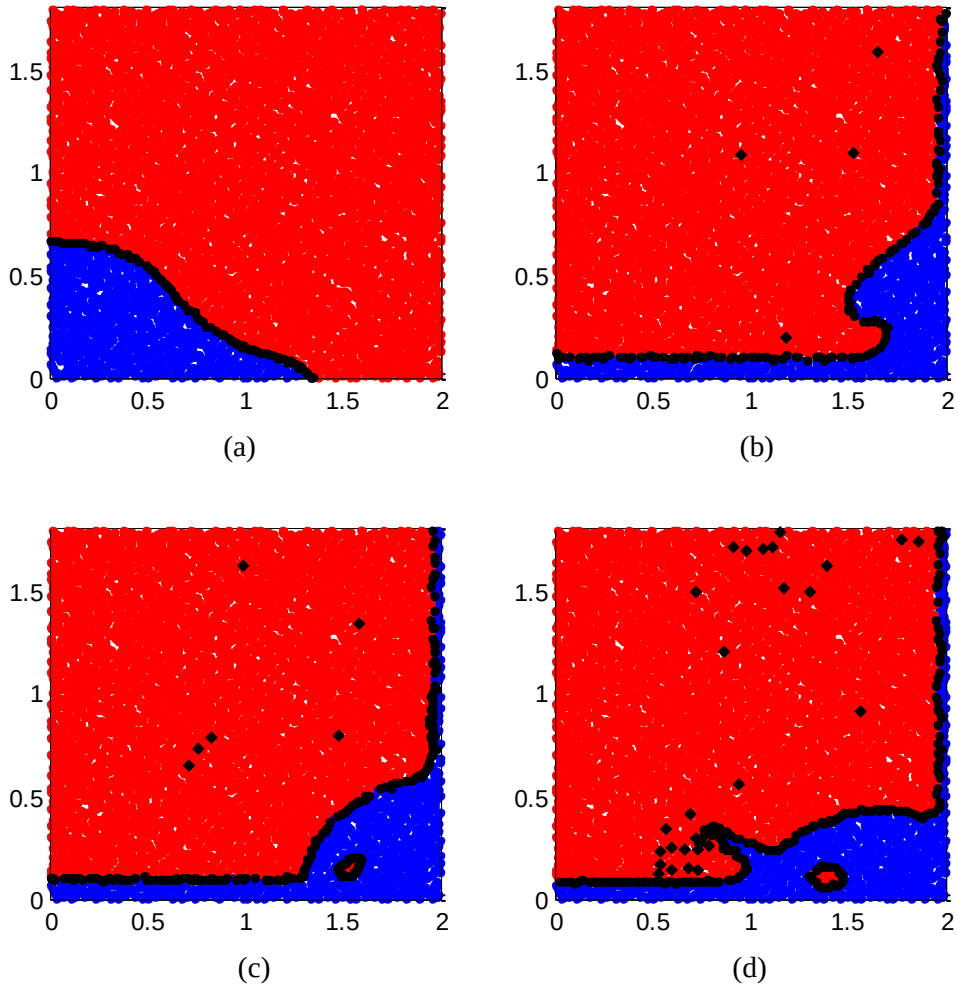


Figure 4.11: Black dots and filled diamonds are interface and isolated particles identified by the ADG technique in four dam breaking snapshots. The heavier fluid with the density of 1.0 of phase 1 is represented by blue dots and the lighter fluid with density of 0.001 of phase 2 is represented by red dots.

A new interface particle identification method for two-phase flows based on the absolute density gradient (ADG) is presented in this chapter. The behaviour of the method is tested for different configurations including these with different level of randomness. It is shown that the accuracy of the method is independent of the density ratio of two phases and also of the average distance between particles. In all the cases tested, the method can correctly pick up almost all the interface and isolated particles as long as the support domain is larger than twice the average distance between particles. The small number of the misidentification happens only near the interface, which does not significantly affect the values of physical variables as they are indeed very close to those on the interface. After identification, the pressure formulations of Eq. (3.54) and Eq. (3.56) will be applied on the interface and isolated particles respectively. For inner particles, Eq. (3.5) is for fixed wall particles and Eq. (3.16) is for fluid ones.

Chapter 5 Algorithms for Solving Pressure Equation

With precise interface and isolated particles identification, the pressure equations derived from interface conditions can be allocated to the correct particles. With the addition of pressure equations for inner fluid particles having the same formulations as in the single phase flow, the pressure equation for the whole computational domain is completed and is ready to be solved by a proper algorithm. It is noted that, for inner particles, the commonly used rigid wall boundary is considered and other boundaries (e.g. symmetric, inlet, outlet etc) will be mentioned if involved. The summary of the final pressure equations is presented in Table 5.1.

Table 5.1: Summary of pressure equations on different types of particles

Particle type	Pressure equation	
Inner fluids	$\left(\int_{\partial\Omega_I} \frac{\boldsymbol{n}\cdot(p_l\nabla\varphi)}{\rho_l} dS - \frac{p_l}{\rho_l}\right) = \int_{\Omega_I} \frac{\boldsymbol{u}_l^{*}\cdot\nabla\varphi}{dt} d\Omega$	Eq. (3.16)
Inner rigid walls	$\boldsymbol{n}\cdot\nabla p_l = \rho_l\left(\boldsymbol{n}\cdot\boldsymbol{g} - \boldsymbol{n}\cdot\dot{\boldsymbol{U}} + \upsilon_l \boldsymbol{n}\cdot\nabla^2\boldsymbol{u}_l\right)$	Eq. (3.5)
Interface	$P(\boldsymbol{r}_0) - \frac{\rho_k\sum_{j=1}^{n} p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0}) + \rho_l\sum_{q=1}^{m} p_k(\boldsymbol{r}_q)\emptyset_k(\boldsymbol{r}_{q0}) - F_r}{\rho_k\sum_{j=1}^{n}\emptyset_l(\boldsymbol{r}_{j0}) + \rho_l\sum_{q=1}^{m}\emptyset_k(\boldsymbol{r}_{q0})} = 0$	Eq. (3.54)
Isolated particles	$P(\boldsymbol{r}_{isp}) - \frac{\sum_{j=1}^{n} p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0}) - F_r}{\sum_{j=1}^{n}\emptyset_l(\boldsymbol{r}_{j0})} = 0$	Eq. (3.56)

5.1 Integrated-1 and coupled-2 method

By adopting MLS method to approximate the pressure or the pressure gradient of inner particles, the final linear pressure equation can be written as $\boldsymbol{K}\cdot\boldsymbol{P} = \boldsymbol{F}$ by combining the discretised version of Eq. (3.16) and Eq. (3.5) (Ma and Zhou, 2009) with Eq. (3.54) and Eq. (3.56). Specifically, it is given as

$$\begin{bmatrix} \boldsymbol{K_{ll}} & \boldsymbol{K_{lI}} & \boldsymbol{0} \\ \boldsymbol{K_{Il}} & \boldsymbol{K_{II}} & \boldsymbol{K_{Ik}} \\ \boldsymbol{0} & \boldsymbol{K_{kI}} & \boldsymbol{K_{kk}} \end{bmatrix} \cdot \begin{Bmatrix} \boldsymbol{P_l} \\ \boldsymbol{P_I} \\ \boldsymbol{P_k} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{F_l} \\ \boldsymbol{F_I} \\ \boldsymbol{F_k} \end{Bmatrix} \tag{5.1}$$

where $l \neq k$ and I denotes the terms associated with interface or isolated particles. It is noted that the elements on the top-right and the lower-left corners are zero. That is because Eq. (3.16) cannot be applied across the interface. In Eq. (5.1) K_{II} is an identity matrix. K_{ll} and K_{kk} represent terms for inner particles within each phase. According to Eq. (3.16) and (3.5) and following Ma and Zhou (2009), their elements can be expressed by

$$\mathrm{K}_{ll,ij} = \frac{1}{\rho_l}\begin{cases} \int_{\partial\Omega_I} \emptyset(\boldsymbol{x}_{ij}) \cdot \boldsymbol{n} \cdot \nabla\varphi dS - \emptyset(\boldsymbol{x}_{ij}) & \text{For inner particles} \\ \boldsymbol{n} \cdot \nabla\emptyset(\boldsymbol{x}_{ij}) & \text{For solid particles} \end{cases},$$

$$F_l = \frac{1}{dt}\begin{cases} \int_{\Omega} \boldsymbol{u}_l{}^* \cdot \nabla\varphi d\Omega & \text{For inner particles} \\ \boldsymbol{n} \cdot (\boldsymbol{u}_l{}^* - \dot{\boldsymbol{U}}) & \text{For solid particles} \end{cases},$$

where $l \in Q_1, Q_2$, i does not denote the interface particles but a particle concerned and j denotes the neighbouring particles of particle i. The elements of $\boldsymbol{K_{lI}}$ are also given by the above equations, which reflects the influence of the interface particles (including the isolated particles) on inner particles through Eq. (3.16).

It is noted that the matrix $\boldsymbol{K}_{Il}$ reflects the influence of inner particles on interface (and isolated) particles through Eq. (3.54) or Eq. (3.56). Due to this, generally $\boldsymbol{K}_{Il} \neq \boldsymbol{K}_{lI}$. Their elements $\mathrm{K}_{Il,ij}$ could be expressed as

$$\mathrm{K}_{Il,ij} = -\frac{\rho_k \emptyset_l(\boldsymbol{r}_{ij}) - F_r}{\rho_l \sum_{j=1}^{n} \emptyset_k(\boldsymbol{r}_{ij}) + \rho_k \sum_{q=1}^{m} \emptyset_l(\boldsymbol{r}_{ij})} .$$

Similarly,

$$\mathrm{K}_{Ik,ij} = -\frac{\rho_l \emptyset_k(\boldsymbol{r}_{ij}) - F_r}{\rho_l \sum_{j=1}^{n} \emptyset_k(\boldsymbol{r}_{ij}) + \rho_k \sum_{q=1}^{m} \emptyset_l(\boldsymbol{r}_{ij})} ,$$

where $m = 0$ or $n = 0$ for isolated particles, and

$\boldsymbol{F}_I = \boldsymbol{0}$.

In order to solve Eq. (5.1), two possible approaches would be considered. One approach is that the pressure is found by solving Eq. (5.1) as one set of equations, as adopted by (Hu & Adams 2009; Khayyer & Gotoh 2013). This is named as 'Integrated-1 Approach (I1APH)' in this thesis for convenience. An alternative approach is that Eq. (5.1) is firstly split into two sets of equations as

$$\boldsymbol{K_{ll}} \cdot \boldsymbol{P_l}^{\boldsymbol{n}} = \boldsymbol{F_l} - \boldsymbol{K_{lI}} \cdot \boldsymbol{P_I}^* , \tag{5.2a}$$

$$\boldsymbol{K_{kk}} \cdot \boldsymbol{P_k}^{\boldsymbol{n}} = \boldsymbol{F_k} - \boldsymbol{K_{kI}} \cdot \boldsymbol{P_I}^* , \tag{5.2b}$$

where the superscript n indicates the current time step, P_I^* are pressure values at interface or isolated particles. P_I^* are estimated by Eq. (3.54) or Eq. (3.56) in which the pressure of the surrounding particles are currently available. Within each time step, iterations may be performed, starting with $P_I^* = P_I^{n-1}$, where the pressure value P_I^{n-1} is obtained at the previous time step at interface or isolated particles. After each iteration, P_I^* will be re-evaluated using the updated pressure. This approach is named as 'Coupled-2 Approach (C2APH)'.

This Coupled-2 Approach is similar to a formulation in which Eq. (3.16) is applied to two phases separately, with the effect of interface and isolated particles taken into account. This is also similar to the decoupled ISPH of Shao (2012) in which the matrices K_{ll} and K_{kk} are decoupled and do not directly include the contribution from the other phase. However, the Coupled-2 Approach (C2APH) here is significantly different from the decoupled ISPH of Shao (2012) in the following aspects. (1) In their approach, there is no explicit requirement on velocity condition at the interface**.** (2) In their approach, the effect of one phase on another phase is taken into account by using the pressure in the previous time step without iteration. Thus the pressure cross the interface in the current time step is not actually ensured to be continuous. (3) Their discretisation of Poisson's equation for pressure is different from what is employed here as they adopted the standard ISPH formulation of approximating the second-order derivative directly. It is noted that the results below will demonstrate that without the velocity condition at the interface, the method can only work for very low density ratios and fails to simulate high density ratios.

5.2 Comparison of two methods

In this section, comparison will be made in the aspect of the accuracy and CPU time of Integrated-1 and Coupled-2 Approaches. For the Coupled-2 Approach, since the pressure of interface particles, as expressed by Eq. (5.2a), is determined by the results of the last time step, the effects of iterations for Eq. (5.2a) within each time step will be first tested for different time step sizes. Then the performance of two algorithms on different density ratios will be examined by comparing the numerical results with theoretical and experimental ones. The cases considered for this purpose are a gravity current flow which is formed by the heavier fluid released into the lighter fluid due to the gravity. For numerical work, the flow is simulated in a 2D tank containing two fluids with the heavier fluid of density of ρ_1 on the left and lighter one of density ρ_2 on the right. The fluids are separated by a partition which will be instantaneously removed to release the heavier fluid

into the lighter. Figure 5.1 illustrates a schematic setup of the numerical tank and initial phase configurations. This problem was broadly addressed by both experimental and theoretical studies and they will be used for pressure solving methods validation in this section.

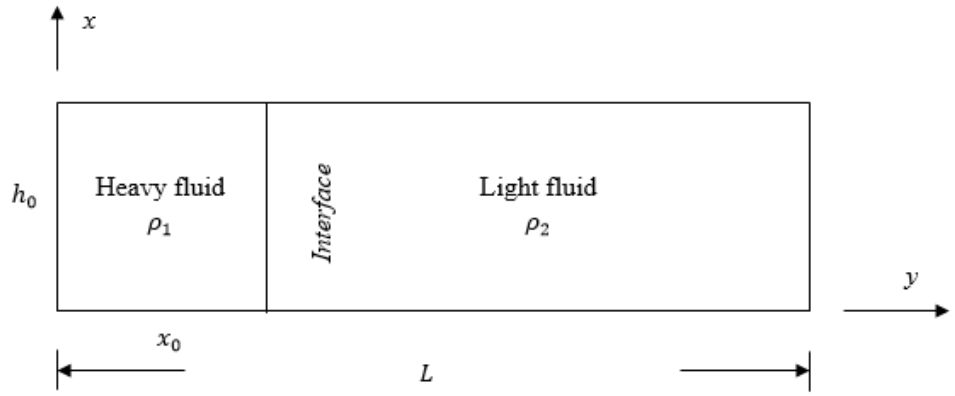


Figure 5.1: Illustration of initial computational setup

Three results of laboratory experiments with different density ratios from Rottman and Simpson (1983), Lowe et al. (2005) and Grobelbauer et al. (1993) will be used. In Rottman and Simpson (1983), the apparatus was a channel with rectangular cross section, 384cm long, 50cm deep and 20.5cm wide. The heavier phase of salt water and the lighter phase of fresh water were divided by a gate placed at a distance x_0 from one of the end walls. When both sections were filled by equal height of two fluids, four tests were conducted with density ratios and dimensions listed in Table 5.2.

Table 5.2: Setups of gravity current flow experiments in Rottman and Simpson (1983)

	Density ratio	$h_0 \ (cm)$	$x_0 \ (cm)$
Test 1	1.01	10	40
Test 2	1.01	10	60
Test 3	1.02	10	50
Test 4	1.04	20	40

It was observed from the experiment that the heavier fluid was displaced forming a gravity current and was propagating towards the end wall. Meanwhile, the lighter fluid was moving oppositely forming a backflow current and a hydraulic drop was generated when it met the wall. The backflow then propagated away from the wall and eventually over took the front of the heavier fluid current. The front positions of the heavier phase

were recorded at several time instants for each test using the dimensionless length of x/x_0 and the dimensionless time of t/t_0 where $t_0 = x_0/\sqrt{g'h_0}$ and $g' = g(\rho_1 - \rho_2)/\rho_2$. With such non-dimentionlization, the propagating of the front was found to be independent of the variables listed in Table 5.2. Thus, the front positions with respect to the time for all the four tests are scatter plotted and data at an early time (before the bore over took the front) can be fitted by a straight line resulting a constant front propagating speed. The same non-dimensionalization will be followed in the numerical simulation and the fitted line from the experiments will be adopted for validations.

The theory for gravity currents used for validation in this section is from Keller and Chyou (1991) in which the full range of density ratio is covered. Based on the observations of their experiments, two possible flow configurations were suggested depending on the density ratio. For low density ratios, close to unity, the oppositely propagating energy-conserving currents are connected by a long wave of expansion and a hydraulic jump, while for high density ratios, the light current is energy-conserving but the heavy current is supposed to have a dissipative front and the connection between two currents is a simple long wave expansion without the hydraulic jump. Although the full range of the density ratio is covered by this theory, the front velocity of the heavy current with density ratio of 1.01, 1.43 and 3.0 accompanying the experimental data will be used to validate two algorithms proposed to solve the discretised pressure equation.

For the simulations by multiphase MLPG_R, the dimensions of the cases are set as $h_0 = 1.0\text{m}$, $x_0/h_0 = 1.2\text{m}$ and $L/h_0 = 8$ which is sufficient to generate a fully developed heavy current without the effect of the end wall. The density ratio $\delta = \rho_1/\rho_2$ is 1.01 for testing the iterations effects for the coupled-2 method. The numbers of particles used along the depth and length are 40 and 320 respectively, whereas the initial distance between particles is 0.025m. Within each time step, the iteration error is calculated by $Er = \frac{\sum_N |p^{i+1} - p^i|}{\sum_N p^{i+1}}$ where i is the number of iterations and the criterion for stopping the iteration is set to be $Er < 10^{-4}$. Time histories of the heavier fluid front position obtained with and without the iteration are illustrated and compared with the linear fitted experiment results (Rottman and Simpson, 1983) in Figure 5.2. (a) to (d), which illustrate the front propagating with decreasing time steps of 0.011s, 0.0096s, 0.008s and 0.0064s respectively. It can be observed that the results obtained by using the iterative approach with all the time steps agree very well with the experiments. However, without the iteration, the results can be significant different from that of the experiments (Figure 5.2a). But the difference goes down with decreasing time steps and can be eliminated when the

time step is sufficiently small (Figure 5.2d). Therefore, for this gravity current flow, the time step of 0.0064s will be adopted to ensure the accuracy of the coupled-2 equations solving algorithm.

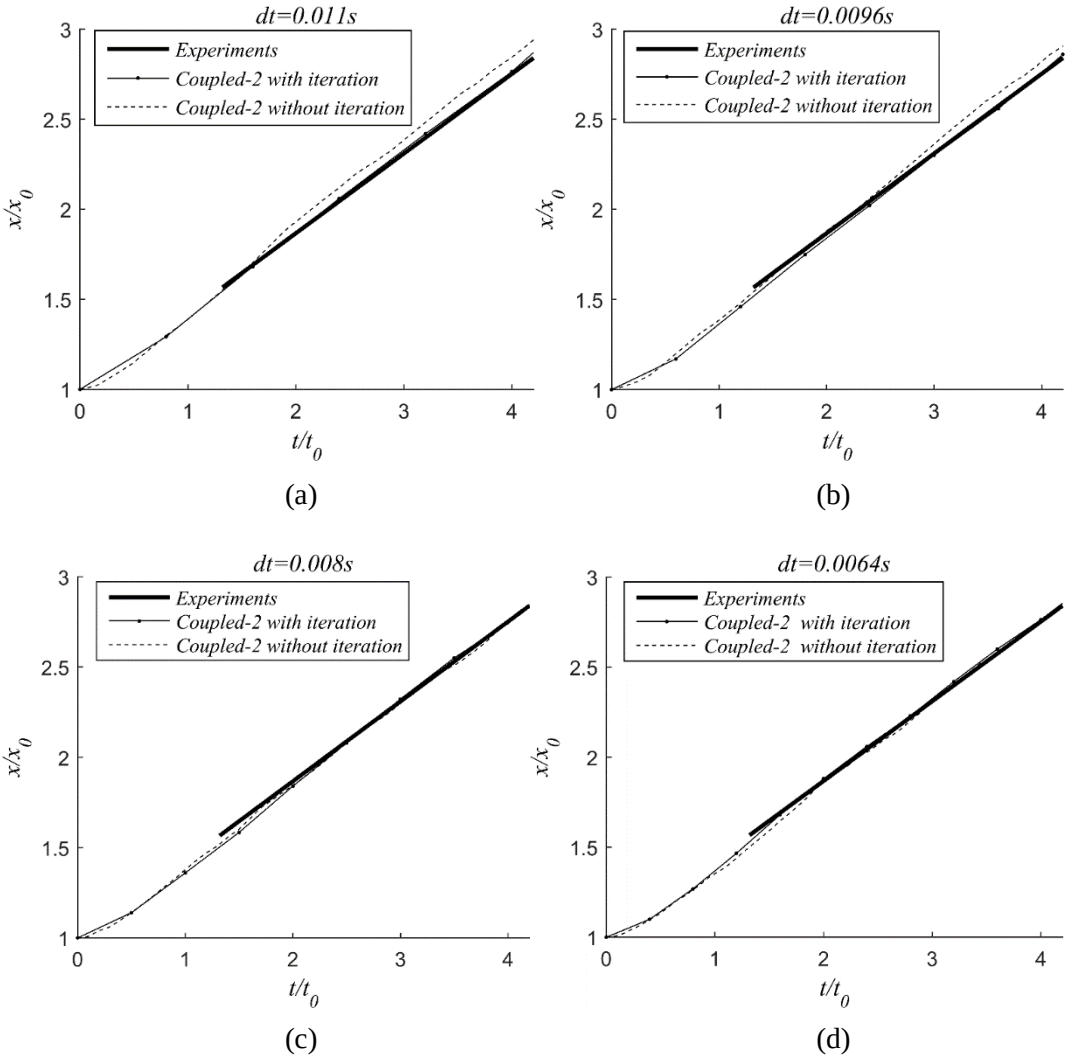


Figure 5.2: Comparison of the front time histories of heavier fluid obtained by using Coupled-2 Approach with or without iteration at time steps of 0.011s, 0.0096s, 0.008s and 0.0064s. (x_0 is initial front position and $t_0 = \mathrm{x}_0/\sqrt{g'h_0}$, $g' = g(\rho_1 - \rho_2)/\rho_2$)

More features of the Coupled-2 Approach will be demonstrated next by comparing its results with those of Integrated-1 Approach. For this purpose, the same case as illustrated by Figure 5.1 will be simulated but with different density ratios of 1.01, 1.43 and 3.0 using a fixed time step of 0.0064s which was tested to be sufficiently small to obtain satisfactory results without iterations. Table 5.3 presents the dimensionless mean front velocity defined by $u^* = u/\sqrt{g'h_0}$ and the CPU time (on Dell OPTIPLEX 790) required

by the two approaches. As for the cases with lower density ratios of $\delta = 1.01$ and 1.43, the mean front velocities obtained by the two approaches are very close to each other and both agree reasonably well with the experiments (Rottman and Simpson, 1983). A theoretical result (Keller and Chyou, 1991) is also given in the table. It is found that the numerical results lie between the experimental and theoretical values. Although both approaches give similar results for the cases, the Coupled-2 Approach costs less CPU time in both cases with the trend that more CPU time may be saved for the higher density ratio. As the density ratio increases to 3.0, the Integrated-1 Approach fails to give convergent results whereas the Coupled-2 Approach keeps working well and yields the front velocity consistent with the experiment (Grobelbauer et al., 1993) and the theory. These tests clearly show that Integrated-1 Approach can only work for the cases with very low density ratio (close to 1) and fails to deal with high density ratio. The Coupled-2 Approach works well for both low and high density ratios. When both approaches work, the Coupled-2 Approach is computationally more efficient.

So far, the multiphase MLPG_R method is fully developed including the pressure formulations for inner, interface and isolated particles by solving pressure Poisson's equation based on a local weak form using the test function of the Rankine source solution, explicitly implementing the continuous pressure and pressure gradient over density and being interpolated by surrounding particles respectively. To solve the discretised pressure equation, decoupled-2 approach is selected as it was justified to cover a wider range of density ratios and to be more efficient than the integrated-1 approach. With the obtained pressure, the velocity and position of each particle is subsequently updated. For flows where there exists large interface deformations or wave breaking, an interface and isolated particle identification technique based on absolute density gradient can be adopted. A flow chart for the whole procedure of multiphase MLPG_R method is shown in Figure 5.3. It should be noted that, since the interface condition considered so far is the continuous pressure and velocity normal to the interface, the applications are restricted to small viscous effects and zero interface tension. In the following chapter, validations of the proposed method will be provided on cases within this scope. Extensions to cover the flows with significant viscous effect and interface tension will be discussed in Chapter 7 and 7.4.

Table 5.3: Comparison of mean velocity of the front and CPU time by applying different methods with density ratio of $\delta = 1.01$, 1.43 and 3.0. (N/A: Not available; N/W: Not working)

Density ratio δ	Methods	$u^* = u/\sqrt{g'h_0}$	CPU time (on Dell OPTIPLEX 790)
1.01	Integrated-1 Approach	0.467	2.02hrs
	Coupled-2 Approach	0.465	1.42hrs
	Experiments (Rottman and Simpson, 1983)	0.440	N/A
	Theory (Keller and Chyou, 1991)	0.493	N/A
1.43	Integrated-1 Approach	0.552	1.58hrs
	Coupled-2 Approach	0.549	0.57hrs
	Experiments (Lowe et al., 2005)	0.480	N/A
	Theory (Keller and Chyou, 1991)	0.597	N/A
3.00	Integrated-1 Approach	N/W	N/W
	Coupled-2 Approach	0.795	1.32hrs
	Experiments (Grobelbauer et al., 1993)	0.659	N/A
	Theory (Keller and Chyou, 1991)	0.872	N/A

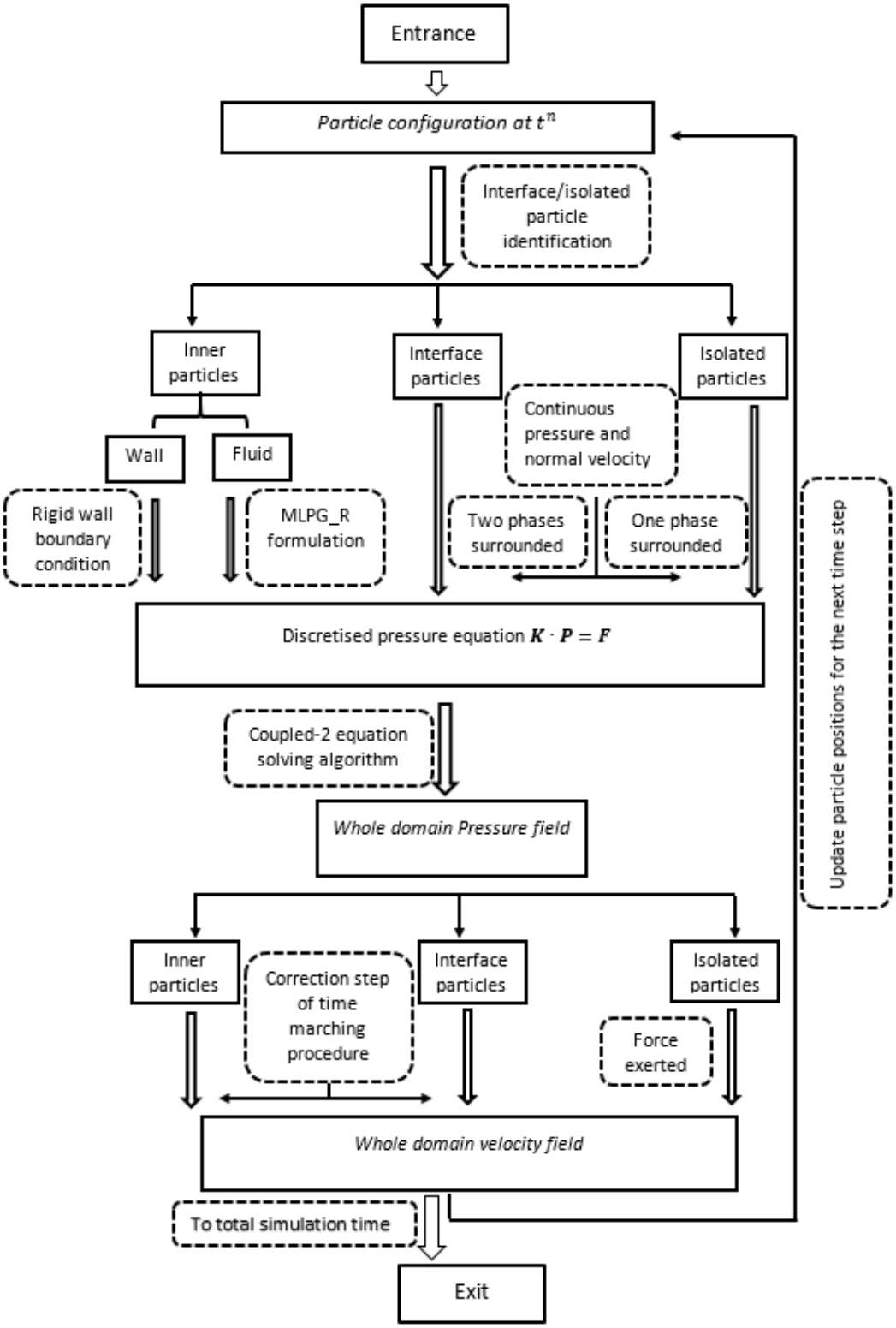


Figure 5.3：Flow chart of the multiphase MLPG_R method procedures

Chapter 6 Results and Validations of Flows with Small Viscosity and Zero Interface Tension

In this section, several cases including both non-breaking and breaking flow situations will be considered to validate the multiphase MLPG_R method which explicitly implements the interface conditions ensuring the continuity of pressure and normal velocity at the interface. For flows with large interface deformations or wave breaking, such as gravity current flow and violent water-air sloshing in section 6.1 and 6.3, the newly proposed interface identification technique described in Chapter 3 will be adopted. As discussed in Chapter 5, the Coupled-2 method will be used to solve the discretised pressure equation to cover a wide range of density ratio from 1.01 to 1000. Validations are conducted with both analytical solutions (i.e. two-phase sloshing with small amplitude) and experimental data (i.e. gravity current flows with different density ratios and water-air violent sloshing).

6.1 Numerical simulation of the gravity current flow

The schematic setup for this case is similar to that in Figure 5.1 for investigating different approaches for solving the pressure equation but the parameters here are different. In this section, the parameters are selected based on an experiment in Rittman and Simpson (1983), in which the filling height of both fluids is $h_0 = 7cm$, the ratio of partition position and tank length to the filling height are $x_0/\,h_0 = 7.14$ and $L/\,h_0 = 28.58$, $g' =$

$47cm/s^2$, and the density ratio is 1.048 (the heavier to the lighter). The apparatus setup and experimental conduction was previously introduced in section 5.2.

In order to verify the effect of time step and the number of nodes, the numerical investigation for convergent property of multiphase MLPG_R method is carried out. For the time step convergent test, Figure 6.1 shows the comparison of the front position propagation of the heavy current for different time steps of $0.84 \times 10^{-3}s$, $1.44 \times 10^{-3}s$ and $2.96 \times 10^{-3}s$ with a fixed particle distance of $0.00175m$ in both directions. It can be seen that little difference is observed between the results of $dt = 0.84 \times 10^{-3}$s and $1.44 \times 10^{-3}s$, whereas the relativly larger time step of $2.96\times 10^{-3}s$ results in a visible but not significant difference. To obtain a more general coefficient for time step indication, Courant number is adopted which is generally defined as $C = udt/dx$ where u is the velocity, dt is the time step and dx is the length interval which is equivalent to particle distance in the meshless method. In the gravity current flow case, u is specified as the maximum velocity in the whole domain which can be regarded as a constant obtained at the front of the heavy current and dx is defined as initial particle distance. In this case, the velocity is obtained by fitting the convergent results (as shown in Figure 6.1) to a straight line giving the gradient of 0.081m/s. By adopting the velocity of the front, the Courant numbers (specified as $C = |u_{max}|dt/dx$) of the convergent time steps become 0.038 and 0.066 corresponding to time steps of $0.84 \times 10^{-3}s$ and $1.44 \times 10^{-3}s$ respectively.

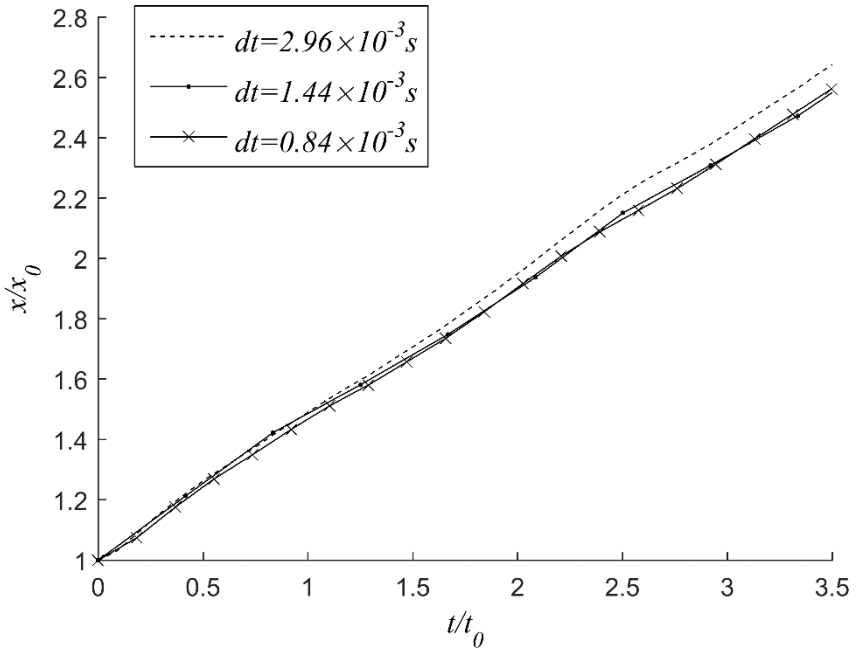


Figure 6.1: Comparison of the leading front time histories of the heavier fluid obtained by using different time steps.

In order to investigate how the results vary with different initial distance between particles, cases with a fixed courant number of 0.038 but different number of particles is considered. To do so, the cases with $dx = h_0/10$, $h_0/20$, $h_0/40$ and $h_0/50$, i.e. 10, 20, 40 and 50 particles are allocated along the depth, corresponding to $dx = 0.007m$, $0.0035m$, $0.00175m$ and $0.0014m$ are investigated. Corresponding results (leading front time histories of heavier fluid) are plotted in Figure 6.2, together with the experimental results from Rittman and Simpson (1983). One can observe that coarse particle distribution ($dx = 0.007m$) results in slower propagation of the head compared to experimental data and the gap is narrowed with refined distribution ($dx = 0.0035m$). The difference becomes invisible and the result is convergent when $dx = 0.00175m$ and $dx = 0.0014m$.

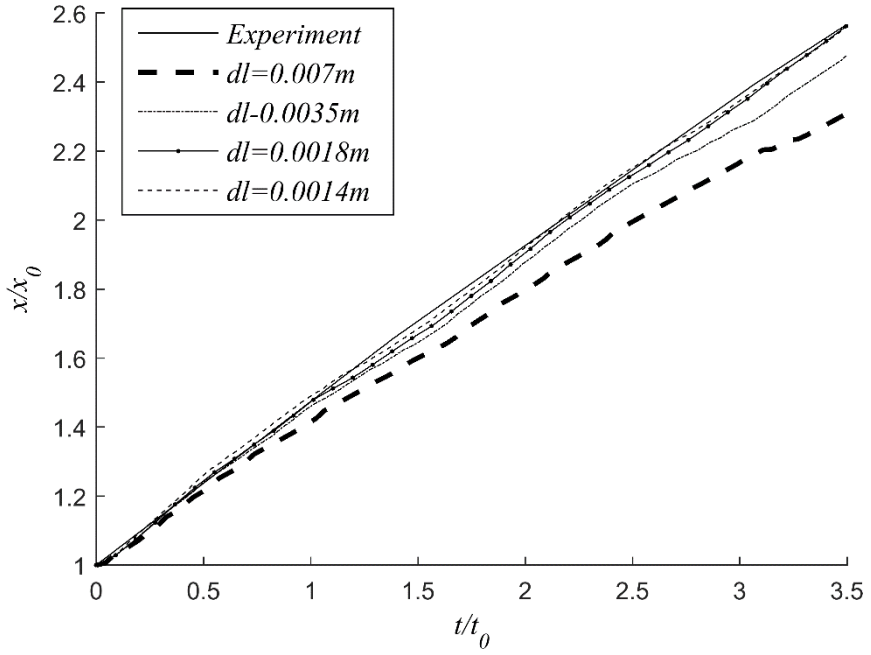
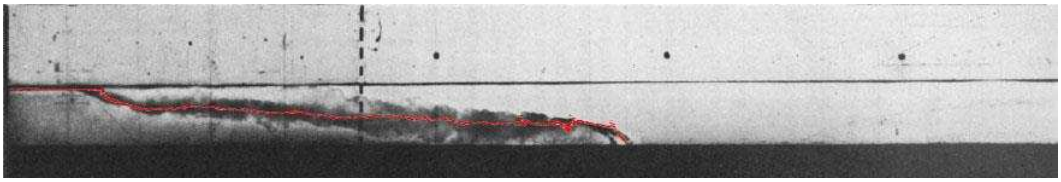


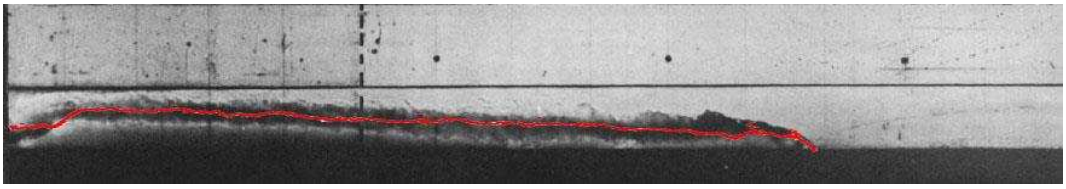
Figure 6.2: Comparison of the leading front time histories of the heavier fluid obtained by using different initial particle distances.

According to the convergence test on the time step and the initial particle distance, $dx = 0.00175m$ and $dt = 0.84 \times 10^{-3}s$ are selected for further validations. Current configurations represented by interfaces at t=4.7s, 7.7s and 10.7s are shown and compared with shadowgraphs of the experiments in (a), (b) and (c) respectively of Figure 6.3. The graphs show the interface from multiphase MLPG_R in red lines with backgrounds of shadowgraphs from experiments of Rittman and Simpson (1983). The dashed line in the shadowgraph marks the position of the partition separating the salt water and fresh water. Besides leading heads, the whole interfaces largely match the experiment as heavier fluid propagates to the right wall and the lighter moves oppositely (Figure 6.3(a)), then the backward current hits the left wall leading to a hydraulic jump (Figure 6.3(b)) and the

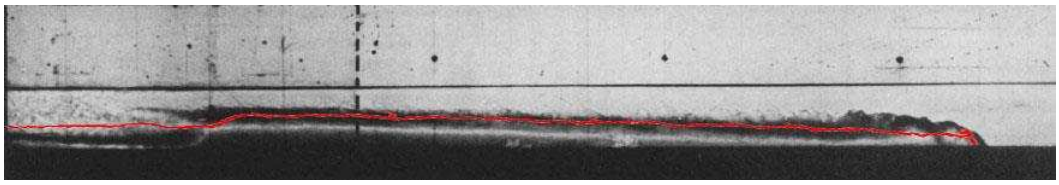
jump continually propagates (Figure 6.3(c)) to eventually overtake the front. Mixing between two fluids behind the front was reported and also could be observed from the shadowgraphs. As multiphase MLPG_R currently models two immiscible phases, such a mixing could not be simulated. But this phenomenon can be explained by circulation zones in Figure 6.4(b) in which velocity vector field is enlarged right after the current front at t=4.7s. In Figure 6.4(a), velocity vectors at the front demonstrate that the heavy current is propagating forward and the light phase right before the current is pushed forward but gradually goes up and backward forming the oppositely propagating light current. Unlike the circulation zone in Figure 6.4(b), independent propagating fields can be observed at the front which hardly lead to mixing. In the same case, pressure fields together with interfaces at three time instants are demonstrated in Figure 6.5. It is noted that the filling depth is non-dimensionalised to be unity and the density of light and heavy fluid are given as 1.0 and 1.048 in the simulation. A smooth pressure distribution can be observed without any unphysical fluctuations and the pressure continuity across the interface proposed as an interface condition is satisfied.



(a)

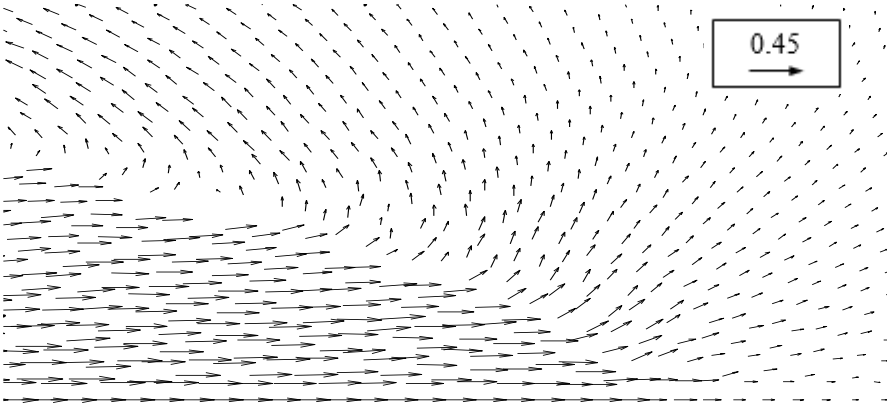


(b)

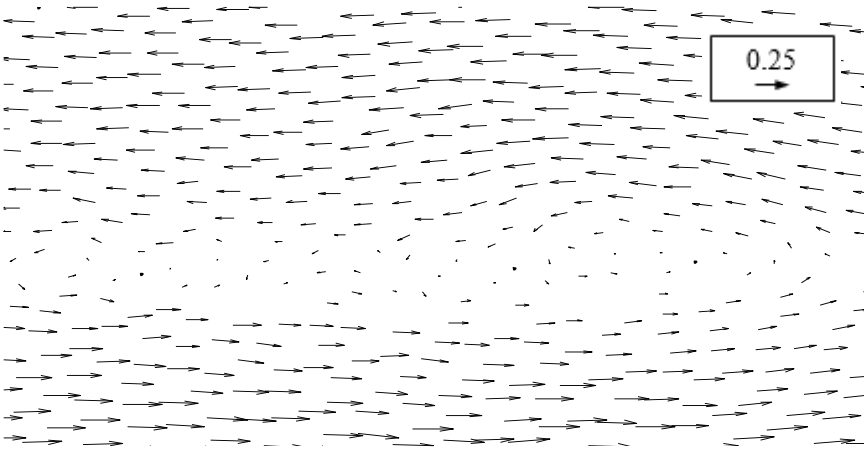


(c)

Figure 6.3: Comparison of interfaces between MLPG_R and shadowgraphs of experiments (Rottman & Simpson 1983) at 4.7s, 7.7s and 10.7s in (a), (b) and (c) respectively.



(a)



(b)

Figure 6.4: Snapshots of velocity (non-dimensional) field in front (a) and right after the front (b) at t=4.7s.

It has been reported (Rottman & Simpson 1983; Grobelbauer et al. 1993; Lowe et al. 2005) that, despite changing the initial setup (e.g. x_0 and h_0), the non-dimensional velocity ($u^* = u/\sqrt{g'h_0}$) of the heavier fluid front largely keeps constant and those constants do not significantly vary with the density ratio if it is not larger than 1.4 (Lowe et al. 2005). More tests with different density ratios (1.048, 1.1 and 1.3) are carried out with a new setup of $h_0 = 0.4m, x_0/h_0 = 1.25$ and $L/h_0 = 8$. To simulate these cases, the initial particle distance is selected as 0.01m by using 40 particles over the depth and the Courant number is chosen as 0.05 to determine the time step.

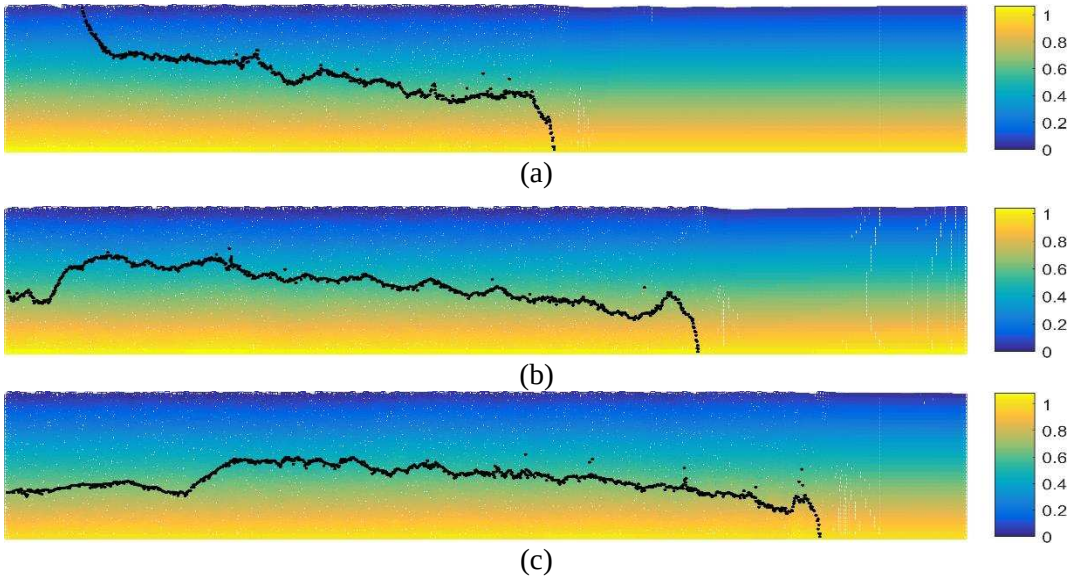


Figure 6.5: Pressure fields of both phases and the interfaces in black dots at 4.7s, 7.7s and 10.7s. The density ratio is 1.048 and the setup follows one experiment in Rottman and Simpson (1983). The length scale is non-dimensionlised by the filling depth.

Figure 6.6 illustrates dimensionless time histories of the leading front position of the heavier fluid with density ratios ranging from 1.01 to 1.3 obtained by various approaches. Scattered squares are experimental data (Rottman & Simpson 1983) for the tests with slightly different density ratios (1.01, 1.02 and 1.04); four curves are obtained by the newly proposed multiphase MLPG_R method for the density ratios of 1.01, 1.048, 1.1 and 1.3; another two curves for the density ratio of 1.3 are obtained by the decoupled incompressible SPH (DISPH) method (Shao 2012) and the MLPG_R method adopting the same interface condition as in DISPH considering pressure continuity only (see Eq. (3.55)). From this figure, one can find that the front position time histories of the experiments and the multiphase MLPG_R method are almost straight lines, implying constant non-dimensional velocity, and correlate very well with each other when the density ratios are less than 1.1. When it increases to 1.3, the result of the multiphase MLPG_R method is still very close to the experimental one, though the difference between them is larger as the density ratio is larger than that in the experiments. This is consistent with the conclusion of Lowe et al. (2005) that the non-dimensional velocity does not vary significantly when the ratio is less than 1.4. One can also find that the result of Shao (2012) for the density ratio of 1.3 is significantly different from the experiments. One of the reasons may be that they did not apply the condition given by Eq. (3.46). To shed some light on this, we also model the case of the density ratio equal to 1.3 without using Eq. (3.46) (i.e. applying Eq. (3.55) rather than Eq.(3.54)). The corresponding results are denoted by 'MLPG_R 1.3 PC' in Figure 6.6. It demonstrates that, for both methods using pressure continuity only, the propagating velocity is not a constant and is

significantly lowered at the beginning but gradually increases. In addition to this, we also find that the separation of two phases may be induced leading to the simulations breaking down without using the condition of continuous normal velocity when the density ratios are higher than 3.0.

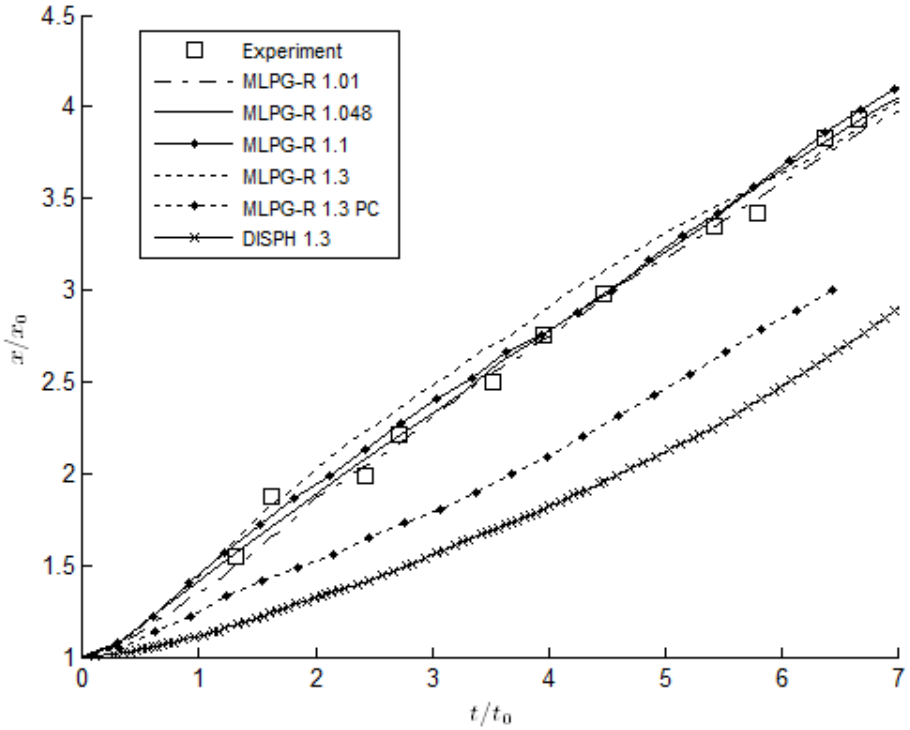


Figure 6.6: Time histories of the leading front of the heavier fluids obtained by the experiments (Rittman & Simpson 1983), by multiphase MLPG_R for the cases with density ratio of 1.01 1.048, 1.1 and 1.3 and by DISPH (Shao 2012) and the MLPG_R method only considering pressure continuity at the interface for the case with the density ratio of 1.3.

6.2 Numerical simulation of two-phase sloshing with small amplitudes

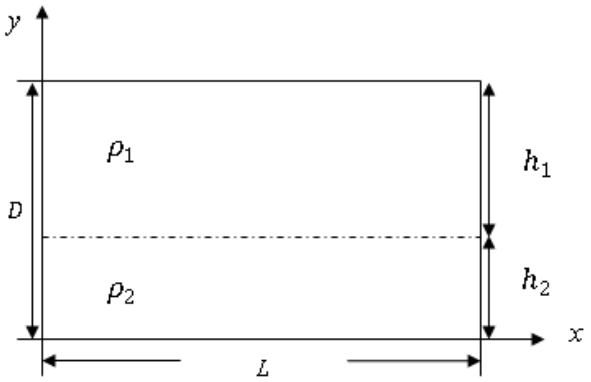


Figure 6.7: Schematic view of tank dimensions and filling fluids for sloshing

The multiphase MLPG_R method is now applied to simulating the sloshing motion of two layered fluids completely filling the 2D tank with the width of *L* and the depth of *D* as shown in Figure 6.7. The heavier fluid with the density of ρ_2 occupies the lower part of the tank up to the depth of h_2 and the lighter fluid with the density of ρ_1 occupies the space with the depth of h_1. A linear analytical solution for two-layer liquid natural sloshing was given by Faltinsen and Timokha (2009) when the motion is small and both fluids are inviscid. This solution is based on continuous pressure through the interface and continuous velocity normal to the interface which is the same as that applied in the MLPG_R method for inviscid fluids. The velocity potentials ($\Phi^{(j)}$) involving two fluids indicated by $j = 1, 2$ are expressed by

$$\Phi^{(j)}(x, y, t) = -a\omega \, sin(\omega t) \, \varphi^{(j)}(x, y) \, , \quad j = 1,2 \tag{6.1}$$

where

$$\varphi^{(1)}(x, y) = C_m f_m(x) \frac{\cosh(\pi m(y - D)/L)}{\cosh(\pi h_1/L)}, \tag{6.2}$$

$$\varphi^{(2)}(x, y) = f_m(x) \frac{\cosh(\pi m y/L)}{\cosh(\pi h_2/L)} \tag{6.3}$$

and

$$\boldsymbol{f_m(x) = cos\left(\frac{\pi m x}{L}\right) \quad m = 1, 2, \ldots,} \tag{6.4}$$

in which σ is the natural frequency, m is the mode number and a is the amplitude of initial disturbance. The pressures in the two fluid domains can by means of the linearized Bernoulli equation be expressed as

$$P_j = \rho_j \left(\frac{\partial \Phi^{(j)}}{\partial t} + gy \right), \quad j = 1,2 \tag{6.5}$$

By satisfying the above mentioned interface conditions, the parameters of A_m, B_m, C_m, the natural frequency σ and the interface elevation ζ are expressed as

$$C_m = -\frac{A_m}{B_m}, \; A_m = tanh\left(\frac{\pi m h_2}{L}\right), \; B_m = tan\, h\left(\frac{\pi m h_1}{L}\right),$$

$$\frac{\omega^2}{g} = \frac{\pi m}{L} \frac{(\rho_2 - \rho_1) A_m B_m}{\rho_2 B_m + \rho_1 A_m} \; , \tag{6.6}$$

and

$$\zeta = aA \frac{\pi}{L} cos(\omega t) f(x) \; . \tag{6.7}$$

In numerical tests, an initial velocity potential was given at the beginning when $\zeta = 0$ across the tank. Alternatively, an initial interface disturbance with zero velocity field by setting $t = 0$ in Eq. (6.7) can be given. However, the initial disturbance may introduce errors due to square particle distribution which cannot precisely match the sinusoidal interface profile. Thus the velocity field at $t = \pi/2\omega$ will be chosen as initial condition in simulations and then the fluids start to slosh without any external excitation. The dimensions of the tank are chosen as D = 2m, $L/D = 1$, the filling ratio of the dense fluid is $h_2/D = 0.5$ and the density ratio is $\rho_2/\rho_1 = 10$. The amplitude a is set to be $0.02m$. As zero viscosity is considered, theoretically there is no energy dissipation and the amplitude of the interface keeps constant at $aA\frac{\pi}{L} f(x)$.

Time step tests are first carried out with an initial particle distance $dx = 0.018m$ with 55 particles allocated along the depth of the dense fluid. For this case, time steps of $dt = 4.5 \times 10^{-3} s, 3.0 \times 10^{-3} s$ and 1.5×10^{-3}s corresponding to the Courant numbers of 0.027, 0.018 and 0.009 are used. The maximum velocity in calculating the Courant number is obtained from Eq. (6.1), which is $|u_{max}| = a\omega\frac{\pi}{L}$. The results of interface elevations at $x = 0.3m$ are shown in Figure 6.8. It is clear that the three tested time steps give similar interface elevation time histories at $x = 0.3m$ while the largest time step

shows more dissipation and damping the interface amplitude. When it is decreased to $dt = 3.0 \times 10^{-3} s$ corresponding to the Courant number of 0.018, it is sufficient to give convergent results and to maintain the constant amplitude.

With fixed courant number of 0.018 to determine the time steps based on the above discussion, different particle numbers of 25, 30, 35, 40, 55 and 80 along the depth of h_2 are then tested, corresponding to initial particle distances of $dx =$ 0.04m, 0.033m, 0.029m, 0.025m,0.018m and 0.013m respectively. Figure 6.9 shows the comparison of interface elevation time histories at $x = 0.3m$ obtained by different particle numbers (only the results of dx = 0.04m, 0.018m and 0.013m are presented for clarity) and analytical solutions by Faltinsen and Timokha (2009). The results indicate that the numerical simulations are converged and give results with little visible difference from the analytical solutions when $dx \leq 0.018m$.

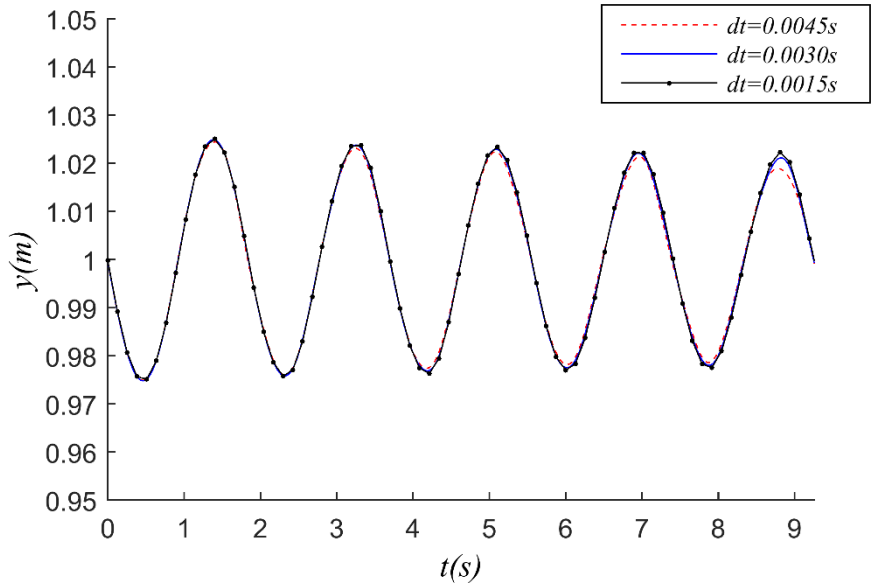


Figure 6.8: Interface elevation at x=0.3m by using different time steps.

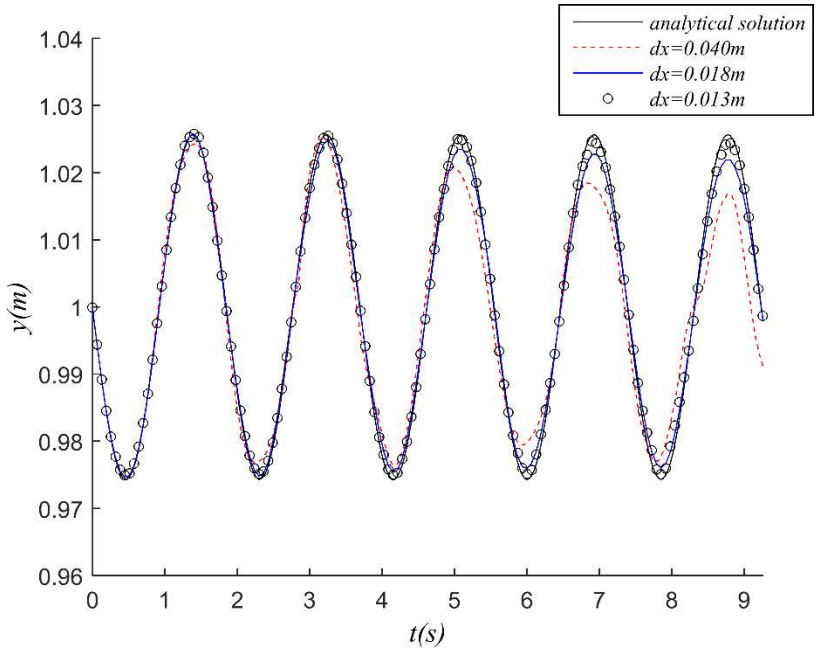


Figure 6.9: Interface elevation at x=0.3m by using different number of particles of MLPG_R method and analytical solutions (Faltinsen & Timokha 2009).

Snapshots of pressure distribution at the first and the third quarter of the first period are shown in Figure 6.10 where the density of the heavy fluid is 1.0 and the light is 0.1. The gravity acceleration is also set to be unity. It shows that the pressure of the two fluids smoothly varies and is continuous at the interface which is illustrated by black dots in Figure 6.10. To further validate the pressure distribution and compare it with the analytical solution, the pressure along the tank depth at $0.3m$ away from the left wall and the right wall is plotted in Figure 6.11. The solid line is obtained by analytical solution showing moderate pressure gradient above the interface and more significant gradient below the interface, which is caused by the density difference of the two fluids. It is clear to observe that the simulated pressure, shown in squares, fits well with the analytical solution and the pressure at the gradient inflection point where the interface located is continuous which is consistent with the interface condition. The regions near the interface points in Figure 6.11(a) and (b) are enlarged to show the interface displacements where the left side ($x = 0.3$) is below the initial filling of h_2 and the right side ($x = 1.7$) is located above it. Both interface displacement and pressure on it fit well with the analytical solution. Figure 6.12 compares the distribution of pressure derivatives over density along the depth given by analytical solution and the two-phase MLPG_R method. One can see that numerical results agree very well with the analytical solution and also that the specific pressure derivative in x-direction is discontinuous while the specific pressure derivative in y-direction is almost continuous. This is consistent with the pressure

condition imposed by Eq. (3.46) with continuous normal pressure derivative over density but discontinuous tangential pressure gradient over density.

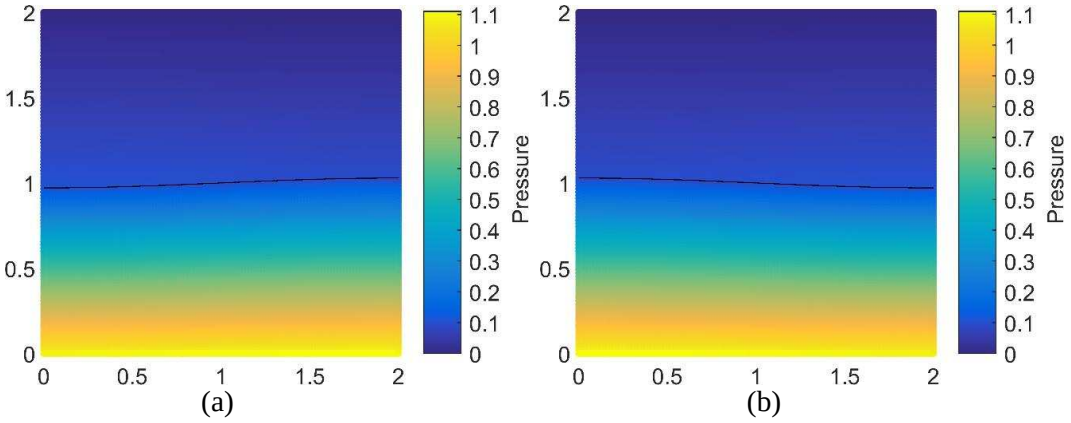


Figure 6.10: Snapshots of pressure distribution at the first quarter (t=0.447s) and the third quarter (t=1.405s) of the first period.

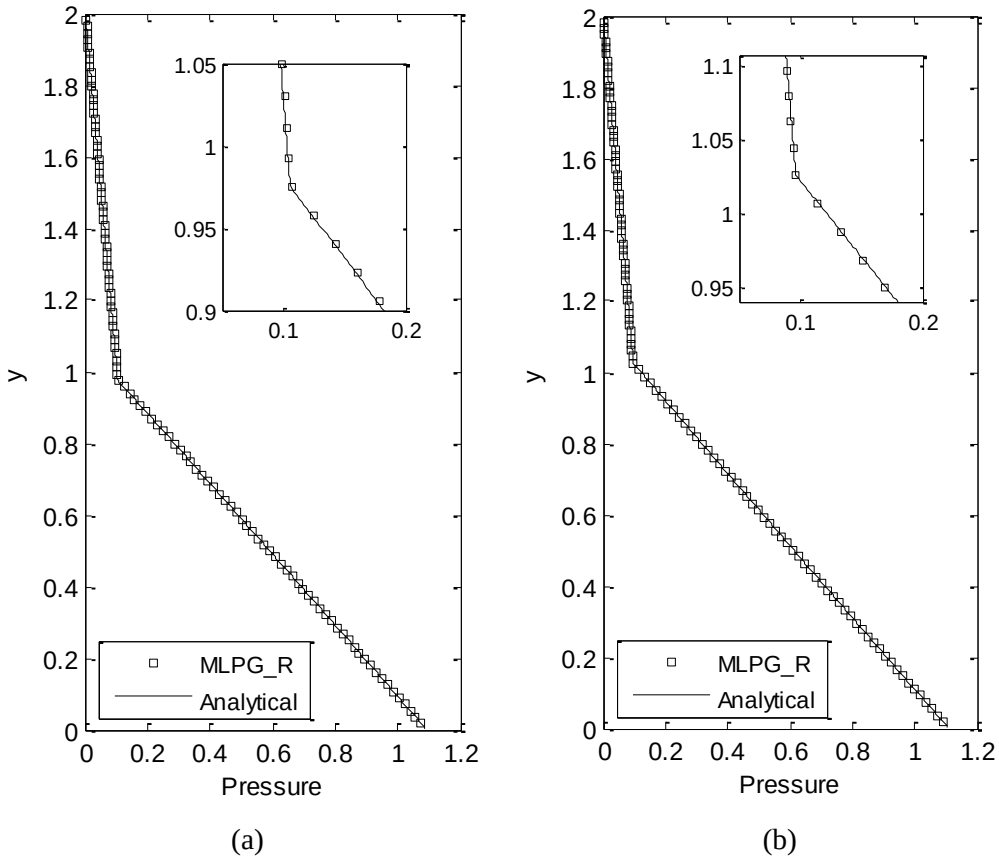
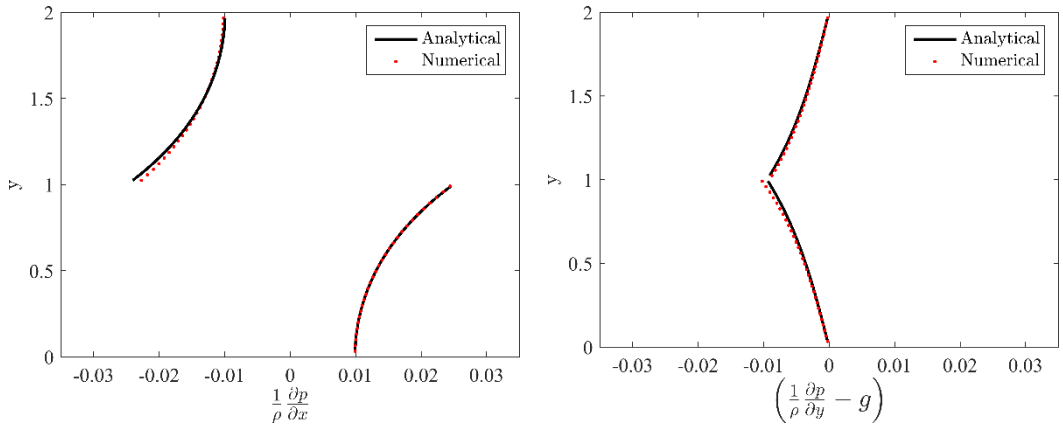
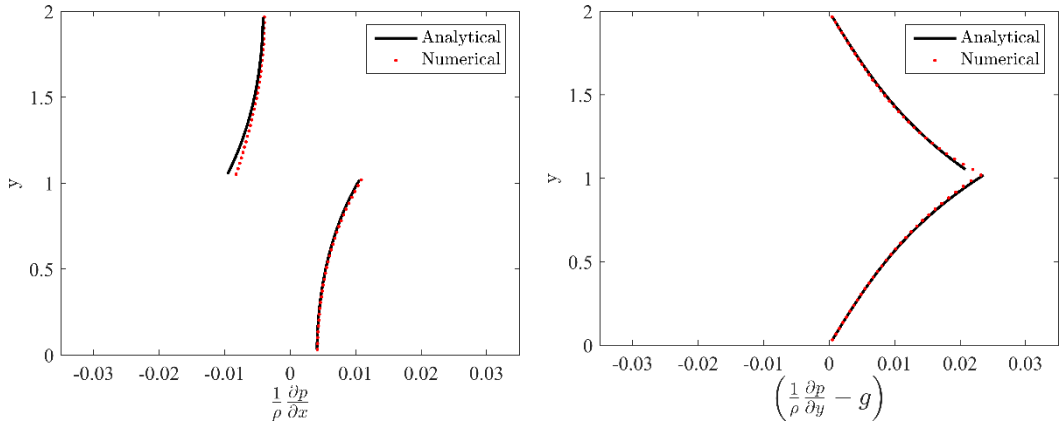


Figure 6.11: Pressure distribution along the depth 0.3m away from the left wall (a) and 0.3m away from the right wall (b) at t=0.447s.



(a)

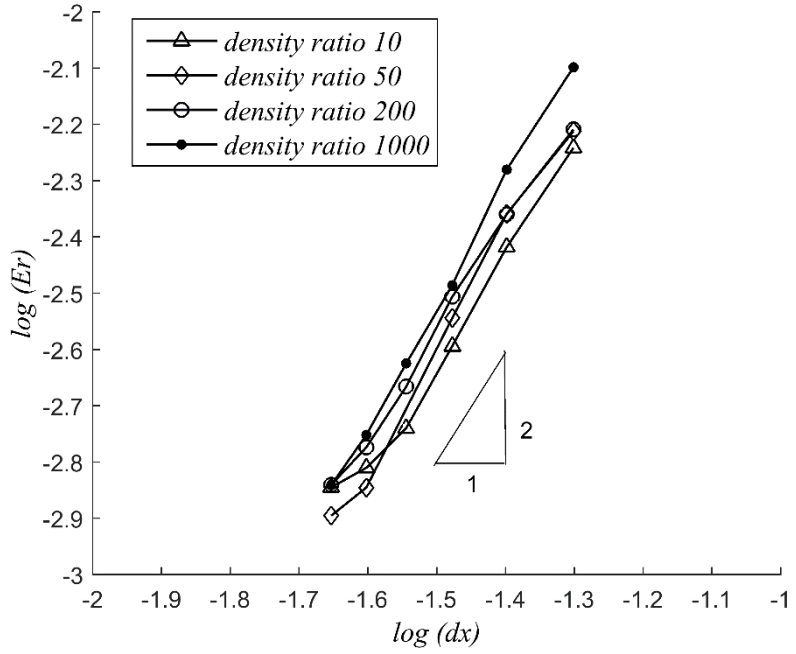


(b)

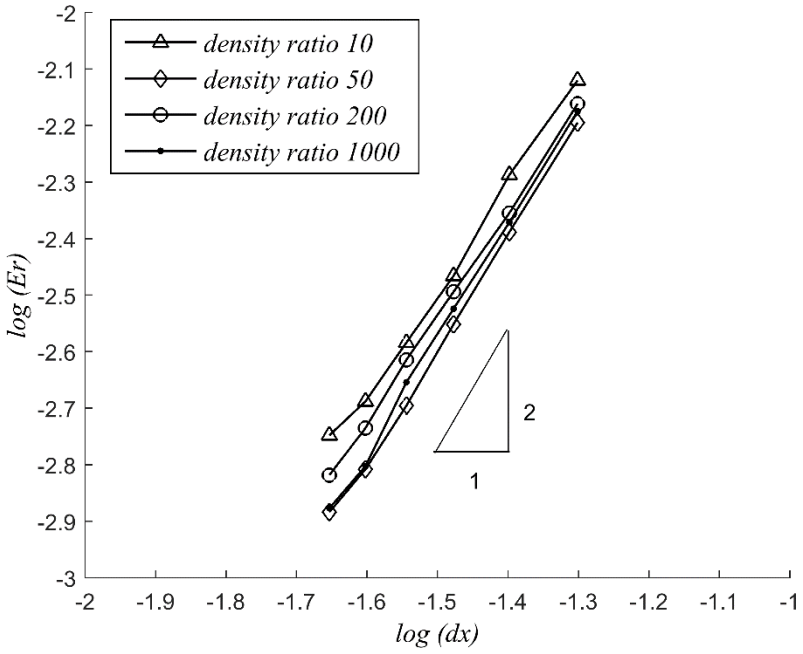
Figure 6.12: Pressure specific derivatives (a)-(b) in $x-$ and $y-$ directions at t=0.75 and x=0.75 and 1.75 with the analytical solution from (Faltinsen & Timokha 2009).

In order to investigate the convergent properties, the error in the results is estimated by comparing it to the analytical solution. The error is estimated by $Er = \sqrt{\sum_{i=1}^{N}\left|\zeta_i - \zeta_{i,a}\right|^2} \Big/ \sqrt{\sum_{i=1}^{N}\left|\zeta_{i,a}\right|^2}$, where $\zeta_{\mathrm{i,a}}$ is the analytical solution of wave elevation at $x = 0.3m$ at i-th time step, ζ_{i} is the corresponding numerical result and N is the total number of time steps during the simulation time. Figure 6.13 shows the errors with respect to the initial distances between particles for two filling ratios of the heavy fluid, which are 0.5 in Figure 6.13(a) and 0.3 in Figure 6.13(b). For either filling ratio, density ratios ranging from 10 to 1000 are tested. One can see that the convergence rates are close to 2 and the impact of the density ratio and filling ratio are negligible. The analytical solution also provides the natural period with specified tank dimension, fluid densities and filling ratios. By fixing the tank dimension as used in simulations, period variations with respect to the filling ratios of the heavy fluid are illustrated by the solid line and the dashed line for the density ratios of 10 and 1000 respectively in Figure 6.14.

Corresponding periods for different filling ratios of 0.1, 0.12, 0.2, 0.3, 0.5, 0.7 and 0.9 for both density ratios are obtained by MLPG_R numerical simulations and good agreement is achieved, which demonstrates the ability of the multiphase model on simulations with high filling ratio such as 0.9 and on low ones like 0.1 as well.



(a)



(b)

Figure 6.13: Numerical errors of simulating density ratio of 10, 50, 200 and 1000 with different particle distances with filling ratio of 0.5(a) and 0.3(b).

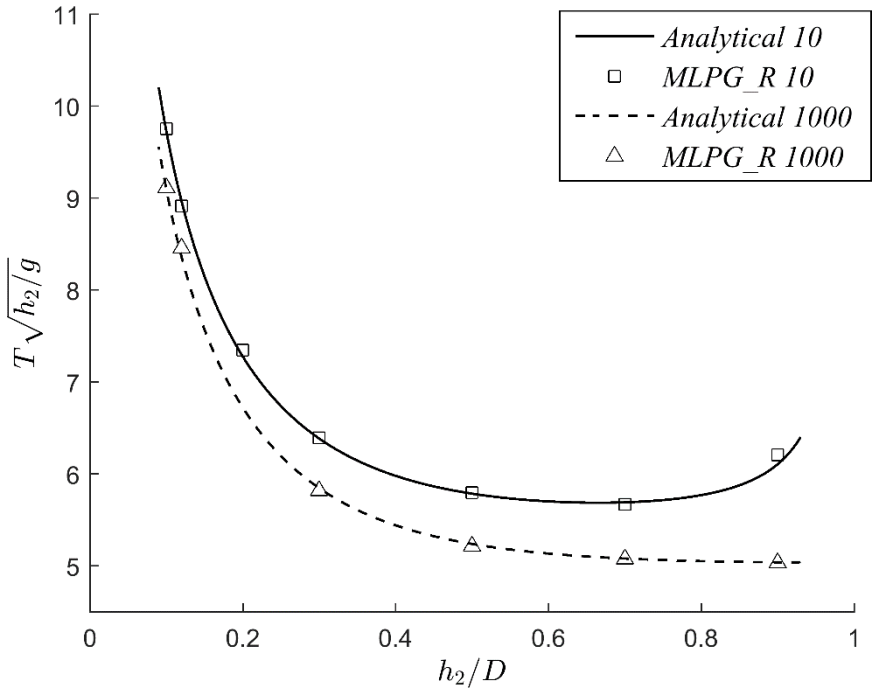


Figure 6.14: Natural periods for different filling ratios of 0.1, 0.12, 0.2, 0.3, 0.5, 0.7 and 0.9. The density ratio is 10 and 1000, respectively.

6.3 Violent sloshing of fluids with large density ratios

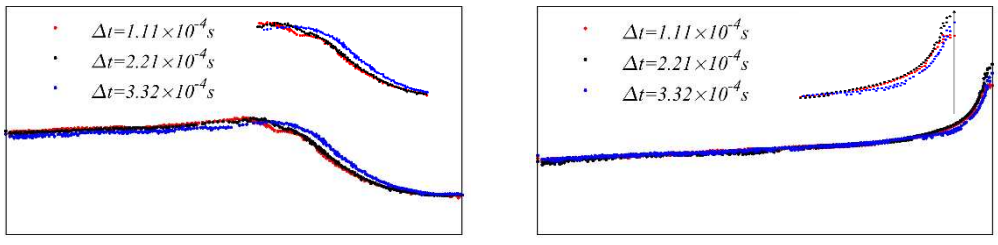
After discussions of the non-violent cases, the multiphase MLPG_R method is further validated using air-water violent sloshing by comparing the results with experimental data provided by Kishev et al. (2006). Such experiments intend to reveal the sloshing occurrence and the forces acting on the tank structure of the partially filled liquid cargo carriers when the ship oscillates in waves. In violent conditions, the force can be large enough to cause local structure deformation or damage. Violent sloshing is a strongly nonlinear problem in which the phenomena like high speed impacts on tank walls, breaking waves, jets, droplets formulation and air bubble entrainment may be involved. Such complicated hydrodynamic phenomena make the theoretical prediction on fluid motions and loads on structures difficult and form strong demands on experiments. But experiments cannot always provide needed information of violent sloshing such as pressure distribution and velocity field in an economical way. Fully nonlinear multiphase MLPG_R model provides a competitive option to obtain such information.

In Kishev et al. (2006), experiments on 2D violent sloshing were conducted in an Plexiglas made tank, 60cm wide, 30cm high, and 10cm deep. The 2D view on height and width plane is the same as that of previously discussed small amplitude sloshing as shown in Figure 6.7. The tank was oscillated in sway with the motion described as $X =$

$a_0 sin(\omega t)$ where a_0 was the sway amplitude and ω was the frequency. On each sidewall of the tank, there were several positions prepared for pressure gauges and the pressure time history obtained by gauge at $x = 0.1m$ on the right hand wall will be used for numerical validation. One high speed camera was used to provide snapshots showing air-water interface profiles.

In violent flow situations, the density ratio is set to be $\rho_2/\rho_1 = 1000$ to simulate air-water sloshing. The same tank dimension and water filling of a set of experiments in Kishev et al. (2006) is set as $h_2 = 0.12m$, $h_2/D = 0.4$ and $h_2/L = 0.2$. The tank is excited by the sway motion, where $a_0 = 0.05m$ and $\omega = 4.83s^{-1}$ (corresponding to the period of 1.3s) are the amplitude and the frequency respectively. In the simulation, the coordinate system is fixed in the moving tank to simplify the solid boundary conditions. One phenomena of the violent sloshing is the formation of water droplets and air bubble penetration which in MLPG_R simulation are illustrated by a few air particles trapped into water and water particles penetrating into air as isolated particles. In that situations, the pressure formulation and identification technique of these particles discussed in section 3.4 and Chapter 4 will be adopted here.

Time step tests of $dt = 1.11 \times 10^{-4}$s, $2.21 \times 10^{-4}s$ and $3.32 \times 10^{-4}s$ with a fixed initial particle distance of 0.004m, i.e. 30 particles along the water depth, are first carried out. By using the maximum velocity given by$|u_{max}| = \sqrt{gh_2}$, the Courant numbers of above time steps correspond to 0.03, 0.06 and 0.09. Water-air interface snapshots at $t = 1.43s$ and $1.56s$ obtained by different time steps are compared in Figure 6.15, with enlarged inserts. The interface differences are mainly at the peak points. But when the time step is shorter than $2.21 \times 10^{-4}s$, the difference is almost invisible. The directing courant numbers of violent sloshing cases are 0.03 and 0.06 corresponding to the time steps of $1.11 \times 10^{-4}s$ and $2.21 \times 10^{-4}s$ respectively.



(a) (b)

Figure 6.15: Snapshots of interfaces obtained by using the time steps of 1.11×10^{-4}s, 2.21×10^{-4}s and 3.32×10^{-4}s at t=1.43s (a) and t=1.56s (b).

By adopting the Courant number of 0.03, initial particle distances of $h_2/10$, $h_2/20$ and $h_2/30$ (corresponding to $dx = 0.012m$, $0.006m$ and $0.004m$) are tested. The interface profiles at different time instants are displayed on the right hand column in Figure 6.16. There is no significant difference between the results corresponding to these three initial particle distances. The left hand column of Figure 6.16 shows the comparison between experiment snapshots (Kishev et al. 2006) and results of smallest particle distance of $dx = 0.004m$. Photos from the experiment were provided for 0.1T, 0.2T, 0.3T and 0.4T (where T is a wave period) without exact time instants. It means that from which period those four instants were picked is not provided. For the simulation snapshots, four snapshots are picked out from the second period, in which the sloshing wave starts to be violent and breaking. The left hand column of Figure 6.16 shows reasonable agreement of wave surfaces between the experiment and the simulation and the right shows little difference between the case with three particle distances. Figure 6.16 also shows that the interface and the isolated particles are almost correctly identified at all the instants. Pressure time histories at $z = 0.83h_2$ on the right wall for three initial particle distances are also compared with experiments and shown in Figure 6.17. It can be observed that the numerical pressure time history is quite smooth and correlated well with the experimental data.

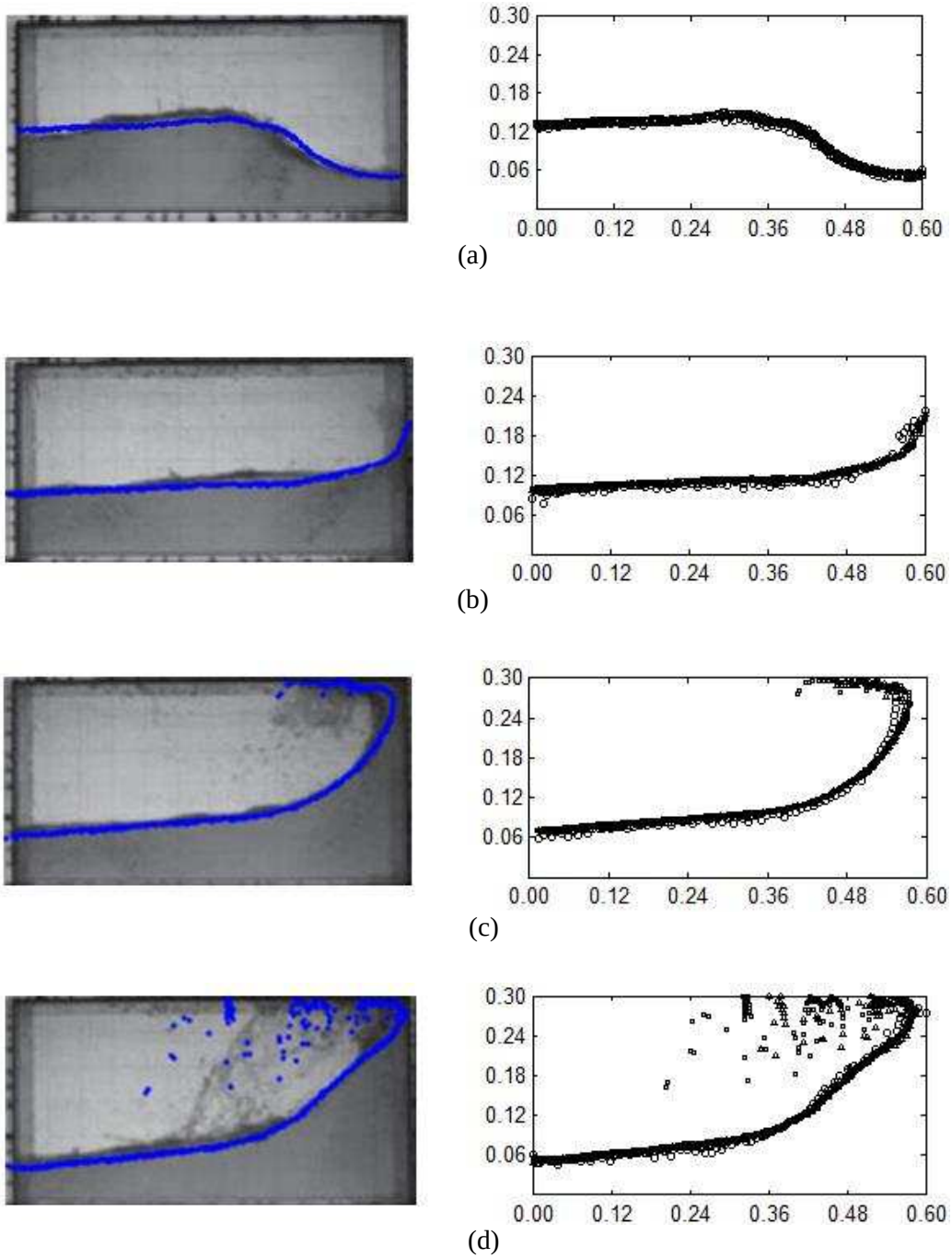
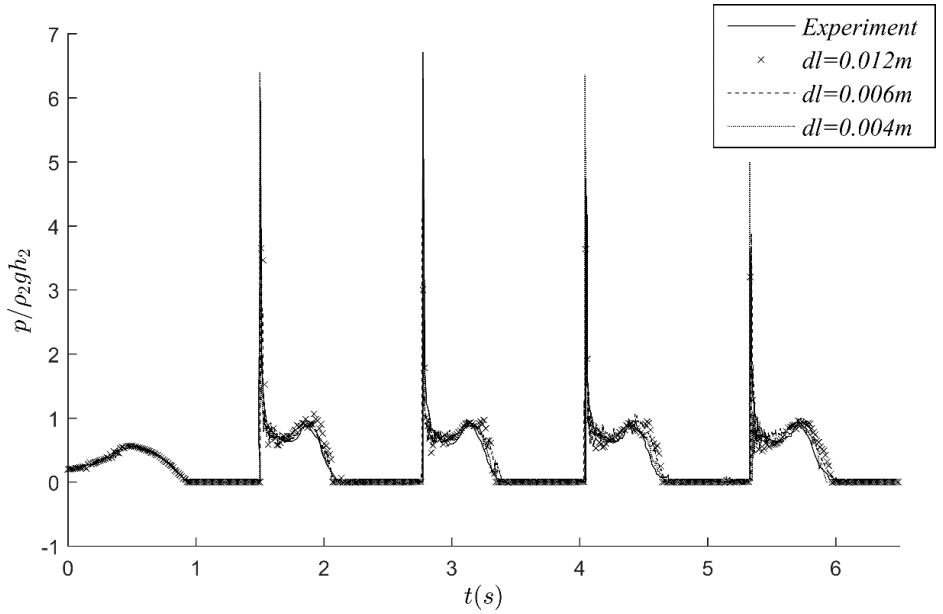
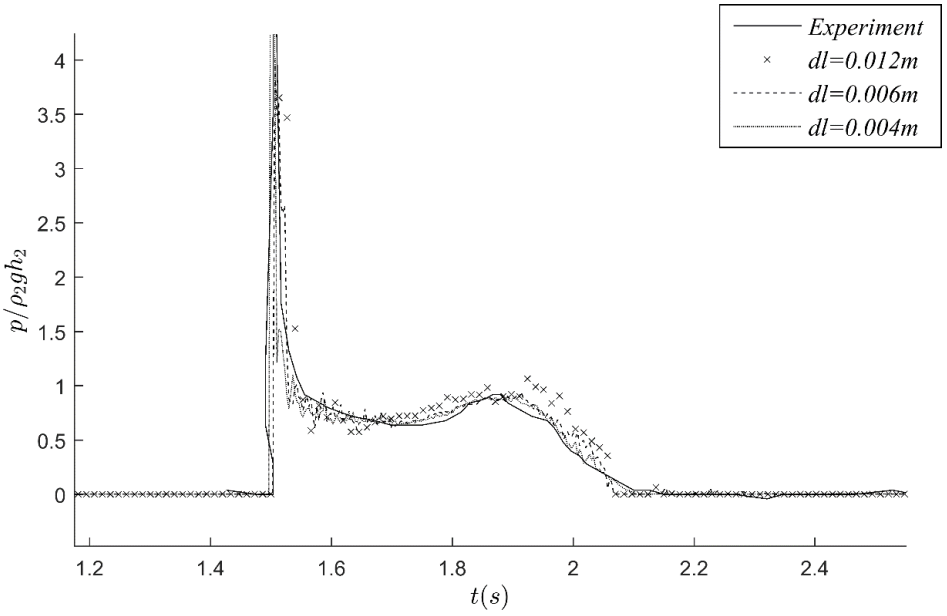


Figure 6.16: Comparison of water-air interface profiles at t=1.43s, 1.56s, 1.69s and 1.82s from (a) to (d). Left column gives experiment photos compared with the simulation of $dx = 0.004m$.Right column gives MLPG_R results with $dx = 0.012m$, 0.006m and 0.004m (snapshots in the shape of circle, rectangle and square respectively).



(a)



(b)

Figure 6.17: Comparison of pressure of $z = 0.83h_2$ on the right vertical wall between experiments and MLPG_R. (a) is the time history for five periods and (b) is the enlargement of the second period.

Chapter 7 Methods for Multiphase Flows with Large Viscosity and Non-zero Interface Tension

In previous chapters, the MLPG_R has been developed to simulate multiphase flows by explicitly implementing the interface conditions of continuous p and $\frac{\partial p}{\rho \partial n}$ (Chapter 3) on the prior identified interface or isolated particles (Chapter 4). As discussed in Chapter 5, the Coupled-2 algorithm is selected to solve the discretised pressure equation to deal with high density ratio cases and to achieve efficiency in computational time. However the current interface conditions are based on the assumptions of neglectable interface tension and small viscous effects. Therefore, the model validations are carried out on the non-breaking and breaking cases satisfying such assumptions in Chapter 6. To further develop the MLPG_R model to simulate more general cases in which interface tension and viscous effects can be more significant, instead of continuity of p and $\frac{\partial p}{\rho \partial n}$ (jump of $\frac{\nabla p}{\rho}$ already existed), a jump on pressure is introduced and the jump on $\frac{\nabla p}{\rho}$ is modified. The jump of pressure on the interface enables the method to deal with the interface tension which ensures the sharp interface satisfying the pressure discontinuity and reduces the spurious currents compared to the widely adopted conventional continuous surface force method (CSF); the modified jump of $\frac{\nabla p}{\rho}$ is introduced to satisfy the interface condition of stress balance when the viscous effects are significant in either phase.

7.1 Interface conditions with pressure jumps

In the previous section, by considering small viscosity and zero interface tension, the continuity in p was implemented. Meanwhile, continuous normal velocity combined with averaged viscous force were also applied at the interface particles. To extend the applications to high surface tension and high viscosity fluids, interface conditions (Brackbill et al. 1992; Sussman et al. 2007) are derived from the continuity condition for velocity in both normal and tangential directions as

$$\boldsymbol{u}_l = \boldsymbol{u}_k \,. \tag{7.1}$$

The jump condition for the pressure caused by viscous stress and interface tension is

$$p_l - p_k = \tau_{l,n} - \tau_{k,n} + \sigma\kappa, \tag{7.2}$$

and the continuity condition for tangential viscous stress is

$$\tau_{l,\tau} = \tau_{k,\tau}, \tag{7.3}$$

where σ is the interface tension coefficient, κ is the local curvature and the sign of the surface tension indicates that phase l is on the convex side. $\tau_{l,n}$ and $\tau_{k,n}$ are normal viscous stresses on two sides and can be expressed by

$$\tau_{\alpha,n} = 2\mu_\alpha n_j \left(\frac{\partial u_j}{\partial n}\right)$$

and

$$\tau_{\alpha,\tau} = \mu_\alpha \left[n_j \left(\frac{\partial u_j}{\partial \tau}\right) + \tau_j \left(\frac{\partial u_j}{\partial n}\right)\right], \quad j = 1,2$$

where $\boldsymbol{n} = (n_1, n_2)$ is the unit normal of the interface towards the fluid with convex interface which is the same as that of then surface tension force, and $\boldsymbol{\tau} = (\tau_1, \tau_2)$ is the unit tangent. According to the above two conditions, p and $\frac{\nabla p}{\rho}$ will be not continuous and jumps will be derived and considered in this section.

The velocity is continuous across the interface, implying that the Lagrangian acceleration is continuous as well. Thus the RHS of Eq. (3.1) needs to be continuous on the interface (Kang et al. 2000) and to satisfy

$$\boldsymbol{F}_l = \boldsymbol{F}_k = \boldsymbol{F} \,, \tag{7.4}$$

where

$$\boldsymbol{F}_\alpha = \boldsymbol{g} - \frac{1}{\rho_\alpha}\nabla p_\alpha + \upsilon_\alpha \nabla^2 \boldsymbol{u}_\alpha \quad \alpha = k, l\,.$$

The expression can be rearranged within each phase and written as

$$\left(\frac{\nabla p}{\rho}\right)_l = \boldsymbol{g} - \boldsymbol{F} + \upsilon_l \nabla^2 \boldsymbol{u}_l\,, \tag{7.5}$$

and

$$\left(\frac{\nabla p}{\rho}\right)_k = \boldsymbol{g} - \boldsymbol{F} + \upsilon_k \nabla^2 \boldsymbol{u}_k\,. \tag{7.6}$$

For low viscosity fluids, $\upsilon_\alpha \nabla^2 \boldsymbol{u}_\alpha$ in Eq. (7.5) and (7.6) are replaced by $(\upsilon_l \nabla^2 \boldsymbol{u}_l + \upsilon_k \nabla^2 \boldsymbol{u}_k)/2$ which ensures the continuity of $\frac{\partial p}{\rho \partial n}$ on the interface. But for high viscosity fluids, averaging the viscous term is no longer applicable and the term needs to be considered separately for each phase. Consequently, the jump of $\frac{\nabla p}{\rho}$ on the interface needs to be modified and expressed as

$$\left(\frac{\nabla p}{\rho}\right)_l - \left(\frac{\nabla p}{\rho}\right)_k = \upsilon_l \nabla^2 \boldsymbol{u}_l - \upsilon_k \nabla^2 \boldsymbol{u}_k\,. \tag{7.7}$$

So far, jumps of pressure and $\frac{\nabla p}{\rho}$ at the interface are derived and expressed by Eq. (7.2) and (7.7) which will be used to formulate the pressure expression of interface particles. The shear stress is estimated with the velocity of the last time step and Eq. (7.3) is used to update the intermediate velocity of the interface particles.

7.2 Pressure formulation of interface particles

Following the section 3.3, the pressure of each phase near the interface particles denoted by $\boldsymbol{r}_0$ is expanded into a Taylor series:

$$\frac{1}{\rho_l}\big(p_l(\boldsymbol{r}) - p_l(\boldsymbol{r}_0)\big) \approx \frac{1}{\rho_l}(\nabla p_l)_{\boldsymbol{r}_0} \cdot (\boldsymbol{r} - \boldsymbol{r}_0) \tag{7.8}$$

and

$$\frac{1}{\rho_k}\big(p_k(\boldsymbol{r}) - p_k(\boldsymbol{r}_0)\big) \approx \frac{1}{\rho_k}(\nabla p_k)_{\boldsymbol{r}_0} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)\,. \tag{7.9}$$

Substituting Eq. (7.5) and (7.6) into Eq. (7.8) and (7.9) and discretising equations in the support domain within each phase yields

$$\frac{1}{\rho_l}\sum_{j=1}^{n}\left(p_l(\boldsymbol{r}_j)-p_l(\boldsymbol{r}_0)\right)\emptyset_l(\boldsymbol{r}_{j0}) = (\boldsymbol{g}-\boldsymbol{F}+\upsilon_l\nabla^2\boldsymbol{u}_l)_{\boldsymbol{r}_0}\cdot\sum_{j=1}^{n}(\boldsymbol{r}_j-\boldsymbol{r}_0)\emptyset_l(\boldsymbol{r}_{j0}), \tag{7.10}$$

$$\frac{1}{\rho_k}\sum_{q=1}^{m}\left(p_k(\boldsymbol{r}_q)-p_k(\boldsymbol{r}_0)\right)\emptyset_k(\boldsymbol{r}_{q0}) = (\boldsymbol{g}-\boldsymbol{F}+\upsilon_k\nabla^2\boldsymbol{u}_k)_{\boldsymbol{r}_0}\cdot\sum_{q=1}^{m}(\boldsymbol{r}_q-\boldsymbol{r}_0)\emptyset_k(\boldsymbol{r}_{q0}), \tag{7.11}$$

where the shape function $\emptyset(\boldsymbol{r})$ is obtained by the moving least square (MLS) algorithm in a support domain containing both phases.

Adding up Eq. (7.10) and (7.11), it follows

$$\begin{aligned}\frac{1}{\rho_l}\sum_{j=1}^{n}p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0})+\frac{1}{\rho_k}\sum_{q=1}^{m}p_k(\boldsymbol{r}_q)\emptyset_k(\boldsymbol{r}_{q0})&\\ -\left[\frac{p_l(\boldsymbol{r}_0)}{\rho_l}\sum_{j=1}^{n}\emptyset_l(\boldsymbol{r}_{j0})+\frac{p_k(\boldsymbol{r}_0)}{\rho_k}\sum_{q=1}^{m}\emptyset_k(\boldsymbol{r}_{q0})\right]&\\ =(\boldsymbol{g}-\boldsymbol{F})_{\boldsymbol{r}_0}\left[\sum_{j=1}^{n}(\boldsymbol{r}_j-\boldsymbol{r}_0)\emptyset_l(\boldsymbol{r}_{j0})+\sum_{q=1}^{m}(\boldsymbol{r}_q-\boldsymbol{r}_0)\emptyset_k(\boldsymbol{r}_{q0})\right]+F_v\ ,&\end{aligned} \tag{7.12}$$

where

$$F_v=(\upsilon_l\nabla^2\boldsymbol{u}_l)_{\boldsymbol{r}_0}\cdot\textstyle\sum_{j=1}^{n}(\boldsymbol{r}_j-\boldsymbol{r}_0)\emptyset_l(\boldsymbol{r}_{j0})+(\upsilon_k\nabla^2\boldsymbol{u}_k)_{\boldsymbol{r}_0}\cdot\sum_{q=1}^{m}(\boldsymbol{r}_q-\boldsymbol{r}_0)\emptyset_k(\boldsymbol{r}_{q0}).$$

Although the first term of RHS of Eq. (7.12) can be omitted when sufficient particles are used, its treatment is similar to that of flows with small/zero viscosity and zero interface tension (see section 3.3) and the term is maintained as $F_r{}'$ computed with parameters obtained from the last time step.

Utilizing Eq. (7.2) and rearranging Eq. (7.12), it follows that

$$p_k(\boldsymbol{r}_0) = \frac{\frac{1}{\rho_l}\sum_{j=1}^{n} p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0}) + \frac{1}{\rho_k}\sum_{q=1}^{m} p_k(\boldsymbol{r}_q)\emptyset_k(\boldsymbol{r}_{q0}) - F_r{}' - F_v - F_{n,k}}{\frac{1}{\rho_l}\sum_{j=1}^{n} \emptyset_l(\boldsymbol{r}_{j0}) + \frac{1}{\rho_k}\sum_{q=1}^{m} \emptyset_k(\boldsymbol{r}_{q0})}, \tag{7.13}$$

$$p_l(\boldsymbol{r}_0) = \frac{\frac{1}{\rho_l}\sum_{j=1}^{n} p_l(\boldsymbol{r}_j)\emptyset_l(\boldsymbol{r}_{j0}) + \frac{1}{\rho_k}\sum_{q=1}^{m} p_k(\boldsymbol{r}_q)\emptyset_k(\boldsymbol{r}_{q0}) - F_r{}' - F_v + F_{n,l}}{\frac{1}{\rho_l}\sum_{j=1}^{n} \emptyset_l(\boldsymbol{r}_{j0}) + \frac{1}{\rho_k}\sum_{q=1}^{m} \emptyset_k(\boldsymbol{r}_{q0})}, \tag{7.14}$$

where

$$F_{n,k} = \frac{1}{\rho_l}\left(\tau_{k,n} - \tau_{l,n} + \sigma\kappa\right)\sum_{j=1}^{n} \emptyset_l(\boldsymbol{r}_{j0}),$$

$$F_{n,l} = \frac{1}{\rho_k}\left(\tau_{l,n} - \tau_{k,n} + \sigma\kappa\right)\sum_{q=1}^{m} \emptyset_k(\boldsymbol{r}_{q0}).$$

The pressures expressed by Eq. (7.13) and (7.14) are implemented on one interface particle and represent the pressure for phase k and l respectively. Compared to interface pressure in section 3.3, the extra terms of F_v and $F_{n,l}/F_{n,k}$ are generated by high viscous force and normal stress jump (includes interface tension) respectively.

7.3 Curvature and surface tension

To achieve an accurate pressure field, the surface tension term $\sigma\kappa$ in Eq. (7.13) and (7.14) needs to be carefully treated. For existing surface tension models, e.g. CSF (Brackbill et al. 1992) and GFM (Fedkiw et al. 1999), calculations are based on the assumption that the location of the interface is known accurately, which is also required by the multiphase MLPG_R method. In mesh based approaches, three methods were proposed to compute curvature. One is estimated from an indicator function c in both VOF and level set approaches and expressed as $\kappa = \nabla \cdot \frac{\nabla c}{|\nabla c|}$ (Hoang et al. 2013; Popinet 2009). c can be a discontinuous fluid distribution function or developed to be smoothed. But such approaches often do not converge with mesh refinement (Desjardins et al. 2008; Cummins et al. 2005). Secondly, a least square approach adopted by Desjardins et al. (2008) in the accurate conservative level set method was tested to be first order convergent. Thirdly, the height function approach implemented in both Level-set (Owkes and Desjardins, 2013) and VOF (Owkes & Desjardins 2013; Owkes & Desjardins 2015) methods achieved a second-order convergence. But extra computations are required to integrate the liquid volume fraction to build the height function for the concerned and neighbouring cells.

For meshless approaches, CSF is also widely adopted for surface tension implementation and the curvature is computed as the divergence of the unit normal using an indicator function (Hu & Adams 2006; Adami et al. 2010) following that in mesh based approaches. But the convergent problem still cannot be avoided. Zhang (2010) proposed a method to locally reconstruct the interface after the interface particle identification which was also adopted in Zhang et al. (2012). In their 2D cases, the interpolation polynomial is used to fit the curve and MLS is used in 3D cases. But interpolation polynomial constructed curve often has oscillations due to irregular interface particle distribution and those oscillations lead to inaccurate first and second derivatives and consequently lead to inaccurate curvatures. To remedy interpolation polynomial induced inaccuracy, MLS interpolation will be used to reconstruct the local interface in MLPG_R method which is less sensitive to the irregular interface particle distribution.

In MLPG_R multiphase model, the interface is automatically traced by identifying interface particles using a numerically computed density gradient (see Chapter 4). Similar with Zhang (2010), a local interface curve will be reconstructed for every interface particles. To ensure a one-valued interface curve, a local coordinate system is first built (as shown in Figure 7.1) with the new origin O' located at the point averaging the coordinates of neighbouring interface particles, i.e., interface particles within the circle, and the y' axis pointing to the concerned particle. Secondly a local curve of the concerned particle will be constructed. Unlike using the polynomial interpolation in Zhang (2010), the second-order MLS technique is used to ensure a smooth curve even with irregularly distributed interface particles. Details of MLS technique is revisited in section 3.2.1 and here the curve reconstruction procedure is presented.

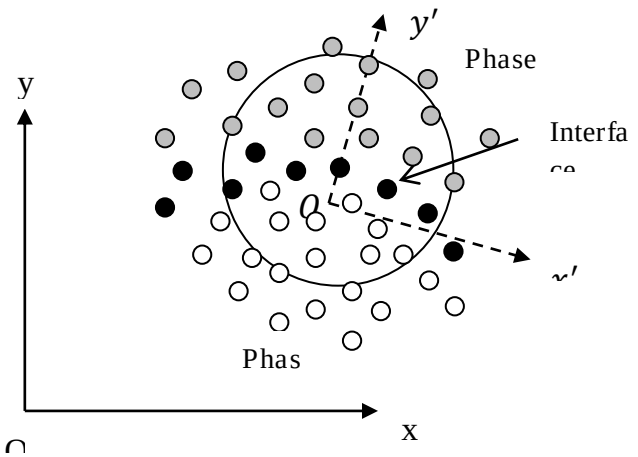


Figure 7.1: Coordinate transformation for local curve construction. xoy is coordinate system for the whole domain and $x'oy'$ is the local coordinate for interface reconstruction for the interface particle passed through by the y' axis

To obtain a second-order curve in the $x'o'y'$ system, the approximant $y'^h(x')$ of $y'(x')$, x ∈ Ω, can be defined by

$$y'^h(x') = \boldsymbol{p}^T(x')\boldsymbol{a}(x') \tag{7.15}$$

where $\boldsymbol{p}^{\mathrm{T}}(x')$ is a complete monomial basis of order m can be written as

$$\boldsymbol{p}^T(x') = [1, x', x'^2], \quad m = 3$$

and $\boldsymbol{a}(x')$ is a vector containing coefficients $a_j(x'), j = 1,2,\dots,m$ and can be expressed as

$$\boldsymbol{a}(x') = \boldsymbol{A}^{-1}(x')\boldsymbol{B}(x')\boldsymbol{y}', \tag{7.16}$$

where $\mathbf{y}'^{\mathrm{T}} = [\mathrm{y}'_1, \mathrm{y}'_2, \dots, \mathrm{y}'_{\mathrm{n}}]$ and the value of $\mathrm{y}_i', i = 1,2,\dots,n$ is from the surrounding interface particle in the support domain. The matrices $\mathbf{A}(x')$ and $\mathbf{B}(x')$ are defined as

$$\boldsymbol{A}(x') = \boldsymbol{P}^T\boldsymbol{W}\boldsymbol{P} = \sum_{i=1}^{n} w_i(x')\,\boldsymbol{p}(x'_i)\boldsymbol{p}^T(x'_i),$$

$$\boldsymbol{B}(x') = \boldsymbol{P}^T\boldsymbol{W} = [w_1(x')\boldsymbol{p}(x'_1), w_2(x')\boldsymbol{p}(x'_2), \dots, w_n(x')\boldsymbol{p}(x'_n)],$$

where the matrices $\boldsymbol{P}$ and $\boldsymbol{W}$ are defined as

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p}^T(x'_1) \\ \boldsymbol{p}^T(x'_2) \\ \cdots \\ \boldsymbol{p}^T(x'_n) \end{bmatrix}, \qquad \mathbf{W} = \begin{bmatrix} \mathrm{w}_1(x') & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots \\ \mathbf{0} & \cdots & \mathrm{w}_{\mathrm{n}}(x') \end{bmatrix}.$$

With solved $\boldsymbol{a}(x')$, the interpolated curve can be expressed as

$$y'^h(x') = a_1(x') + a_2(x')x' + a_3(x')x'^2. \tag{7.17}$$

The curvature and the normal vector of the interface can be calculated from the interpolated curve. The curvature κ is given by

$$\kappa = \frac{\left|d^2y'^h/dx'^2\right|}{[1 + (dy'^h/dx')^2]^{3/2}}. \tag{7.18}$$

The normal vector with the same direction of the surface tension pointing to the concave fluid in the local coordinate system is taken as

$$\boldsymbol{n'} = \begin{cases} \left(\dfrac{\boldsymbol{dy'^h}}{\boldsymbol{dx'}}, \boldsymbol{-1}\right), & \boldsymbol{for\ \ d^2y'^h/dx'^2 < 0} \\ \left(\boldsymbol{-\dfrac{dy'^h}{dx'}}, \boldsymbol{1}\right), & \boldsymbol{for\ \ d^2y'^h/dx'^2 > 0} \end{cases} . \tag{7.19}$$

It is noted that the curvature does not change with the transformation of the coordinate system but the normal vector needs to be converted back to the original xoy system shown asbxl

$$\boldsymbol{n} = \begin{cases} \left\langle \dfrac{dy'^h}{dx'} cos\alpha + sin\alpha, \dfrac{dy'^h}{dx'} sin\alpha - cos\alpha \right\rangle, \\ \qquad for\ \ d^2y'^h/dx'^2 < 0 \\ \left\langle -\dfrac{dy'^h}{dx'} cos\alpha - sin\alpha, -\dfrac{dy'^h}{dx'} sin\alpha + cos\alpha \right\rangle, \\ \qquad for\ \ d^2y'^h/dx'^2 > 0 \end{cases} \tag{7.20}$$

where α is rotation of the local coordinate from the original coordinate.

The curvature obtained by the above technique is tested on different types of curves including a circle, $x^2 + y^2 = 1$, an ellipse, $\left(\frac{x}{2}\right)^2 + y^2 = 1$, and a sinusoidal curve, $y = sin(x)$. Particle distances of 0.167, 0.1, 0.063, 0.05, 0.033, 0.02 and 0.013 are tested for the three curves and the L^2 relative errors defined by $E_{L^2} = \sqrt{\frac{\sum_{i=1}^{N}(\kappa_i - \kappa_{ia})^2}{\sum_{i=1}^{N} {\kappa_{ia}}^2}}$ are plotted in Figure 7.2, where κ_{ia} is the analytical curvature on the curve particle and κ_i the corresponding numerical curvature. One can observe that second-order convergence is achieved by the local curve fitting with second-order MLS method.

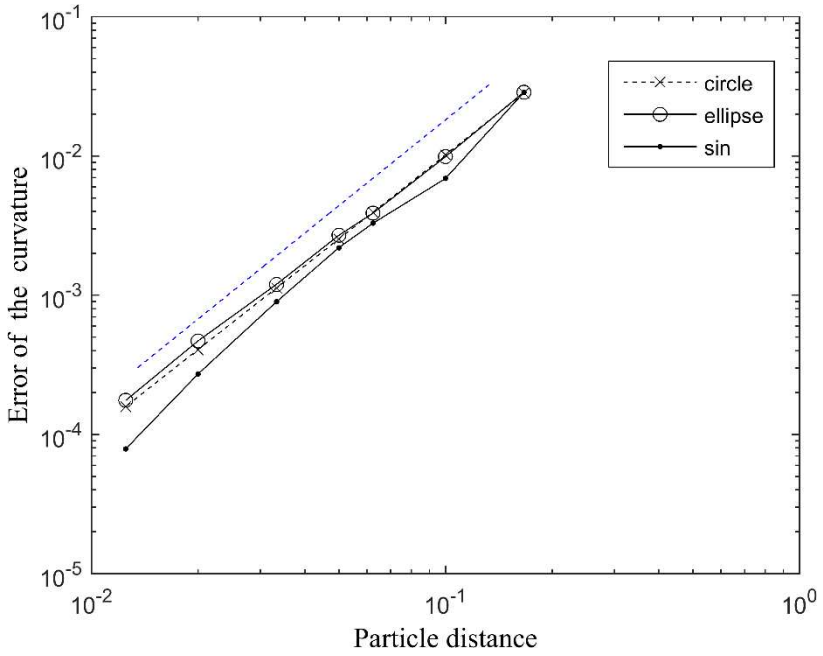


Figure 7.2: Convergence of L^2 error with different particle distance for the circle, ellipse and sine-curve test case. The dashed line in blue indicates the second order convergence.

7.4 Results and Validations of Flows with Interface Tension and Large Viscosity

In this section, several cases will be simulated to show the sharp pressure jump on the interface caused by interface tension and how the accurate curvature and balanced interface force prevents spurious currents. With the modified jump of $\frac{\nabla p}{\rho}$ (shown by Eq. (7.7)) derived from the velocity continuity at the interface, fluids with large viscosity can be accurately simulated.

7.4.1 Numerical simulation of square-droplet deformation

In this zero gravity case, an initially square shaped fluid with density of ρ_1 and dynamic viscosity of μ_1 is located in the centre of a larger square as shown in Figure 7.3. The density of the fluid surrounding the inner square is ρ_2 and the viscosity is μ_2. Due to unbalanced surface tension force, capillary waves are induced to oscillate the inner fluid about its equilibrium shape which is a circle based on Laplace law (Landau & Lifshitz 1987). Viscous effects damp the oscillation and lead the inner fluid oscillate to its final equilibrium shape.

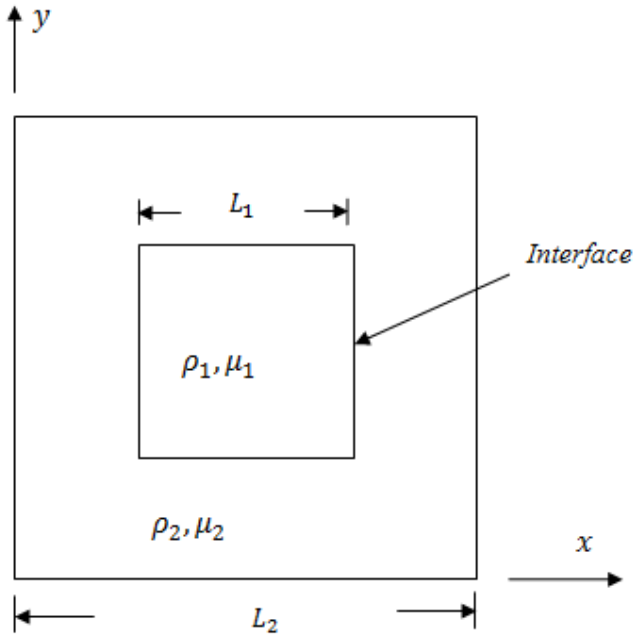


Figure 7.3: Set up of square-droplet deformation test

This case has been widely tested to validate surface tension models and their curvature estimations (Brackbill et al. 1992; Adami et al. 2010; Zhang et al. 2012; Hoang et al. 2013). Main problems shared by those simulations are that interface covers several cells (in the mesh based methods) or particle layers (in meshless method) within which curvature and surface tension are smoothed. Such smoothed interface tension force does not strictly satisfy the interface condition stated by Eq. (7.1) and (7.2). In addition, the pressure drop on the interface is smeared rather than sharp as stated in interface boundary conditions, leading to relatively high spurious currents that occur when the droplet reaches its equilibrium. With the MLPG_R surface tension model, a sharp pressure jump within one interface particle layer is achieved and the spurious currents are significantly relieved.

The test follows the settings in Hoang et al. (2013) and all the parameters are non-dimensionlised. The density of the surrounding fluid is $\rho_2 = 3546$, viscosity is $\mu_2 = 1.0$, density ratio of surrounding and inner fluids is $\rho_2/\rho_1 = 2$, viscosity ratio is $\mu_2/\mu_1 = 0.4$ and the interface tension is $\sigma = 1.0$. The domain size $L_2 = 2L_1 = 4$ and the inner square is located in the centre of the outer square. When the inner drop reaches its equilibrium state, the shape is transformed to a circle remaining the same area as the initial square. According to Laplace's law, the pressure inside and outside the circle are two constants having a jump at the circle. By plotting the pressure at $y = 0.5$ from the centre of the circle ($x = 2.0$) to the edge of the outer square ($x = 4.0$), Figure 7.4 illustrates the higher

constant pressure inside the inner circle and the lower constant pressure outside the circle. The translation between the two constants is completed on the interface particle. One can also observe that the simulations using 40, 60 and 80 particles along the outside square edge largely give similar results which are close to the analytical solution. From the enlarged window in Figure 7.4, good convergence can be observed when 80 particles are used. The reason of such sharp pressure jump is that in our newly proposed surface tension model, the pressure jump condition is explicitly satisfied. By storing two pressure values for two phases on one interface particle, the pressure jump over an infinitely thin interface is achieved. Similar simulations were conducted by SPH method in Zainali et al. (2013), the pressure across the interface gradually decreased from the inside to the outside of the drop which loses the sharp pressure jump stated in Laplace law.

By recording the maximum velocity in the whole domain, Figure 7.5 shows that after a long time, the drop reaches its equilibrium and due to the energy dissipation of viscosity the maximum velocity is gradually reduced and becomes a stable but non-zero value which is caused by numerical errors. This non-zero stable velocity is called the spurious current. By checking the maximum velocity dissipation, not only the convergence is achieved when 80 particles are used, compared to conventional CSF model, but also the spurious current is significantly reduced and stabilized by the new interface tension treatment.

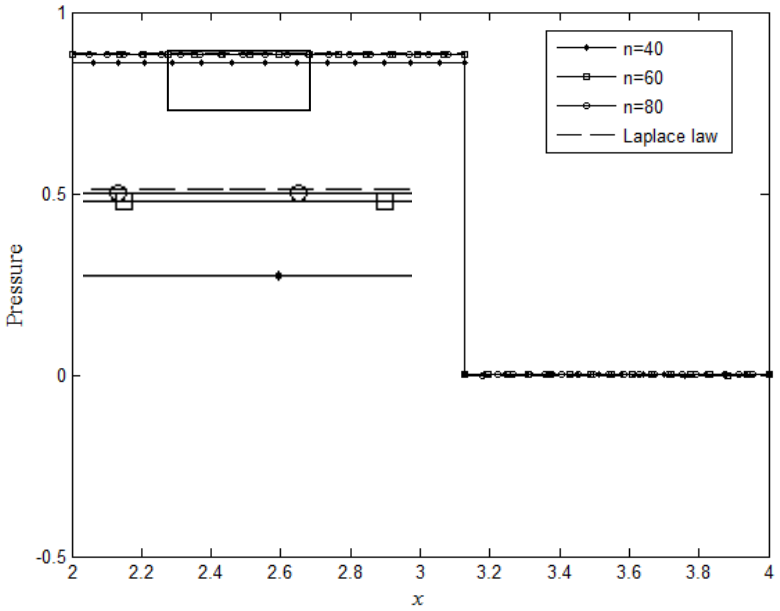


Figure 7.4: Pressures measured from the centre of the circle ($x = 2.0, y = 2.0$) to the edge of the outer square ($x = 4.0, y = 2.0$). Pressure drop occurs at the interface of two fluids at $x = 3.13$.

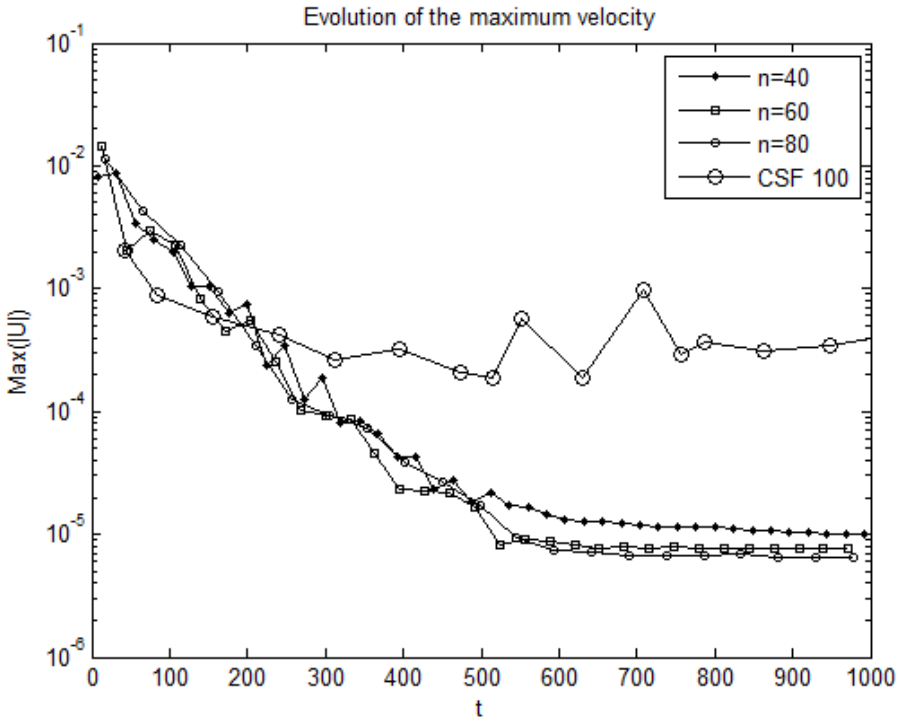
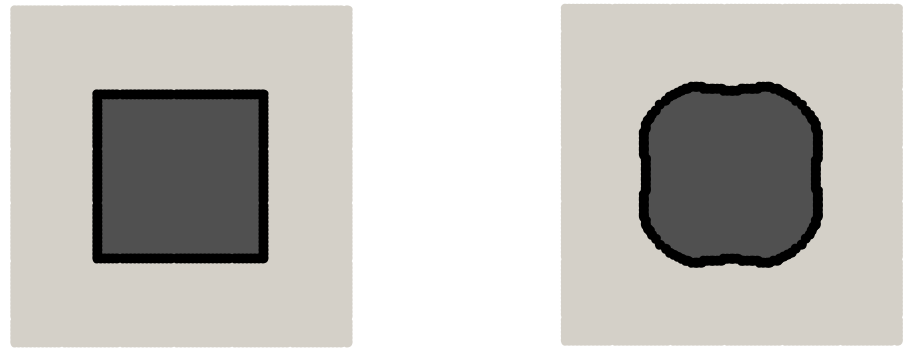
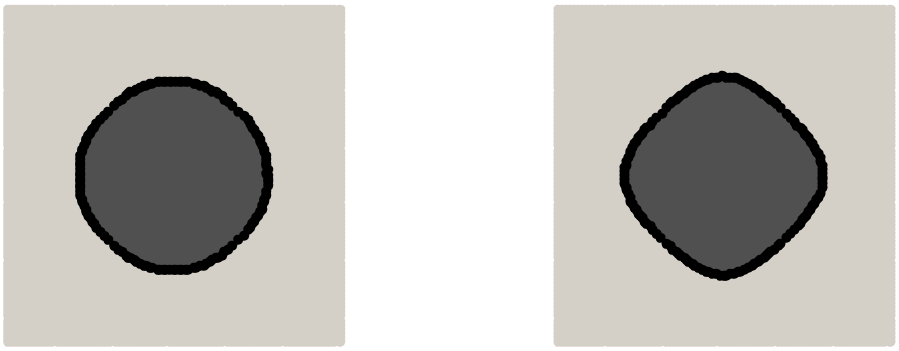


Figure 7.5: Time history of maximum velocity with different particle numbers along the outside square edge. The large circle solid line is from Hoang et al. (2013) adopting CSF model with cell number of 100.

By using 80 particles along the outside square edge, the shape change at a sequence of time instants, $t = 0, 12, 24\ and\ 36$ are shown in Figure 7.6. The initially square shape of the drop results in strong interface tension force at high-curvature corners, forcing the drop to be oscillated. The corners of the drop gets more smooth and eventually becomes a circle as shown in Figure 7.7(a) whose curvature is equivalent at every point. Correspondingly, the pressures inside and outside the drop become two constants with a sudden jump at the circle which is illustrated in Figure 7.7(b).

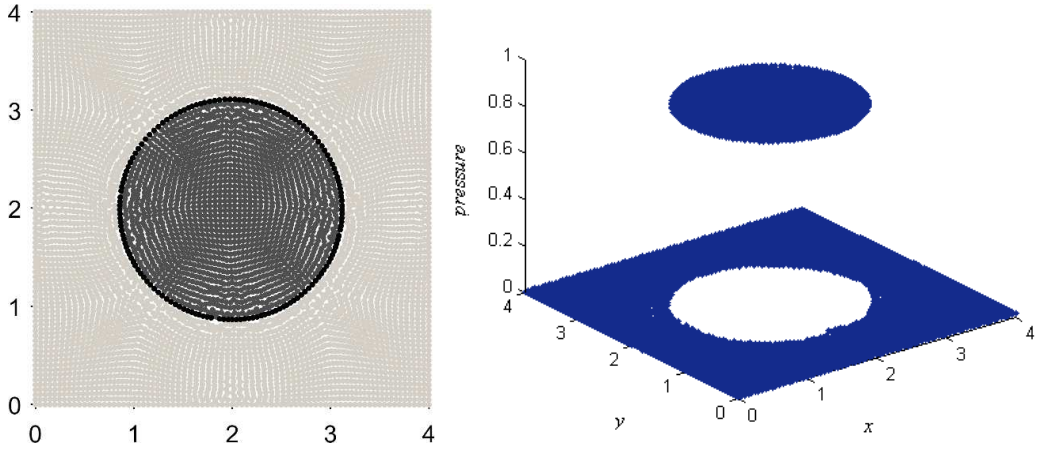


(a) (b)



(c) (d)

Figure 7.6: Drop shapes at $t = 0, 12, 24$ and 36 in plots from (a) to (d) respectively. Interface, inner and surrounding particles are in black, dark grey and light grey respectively.



(a) (b)

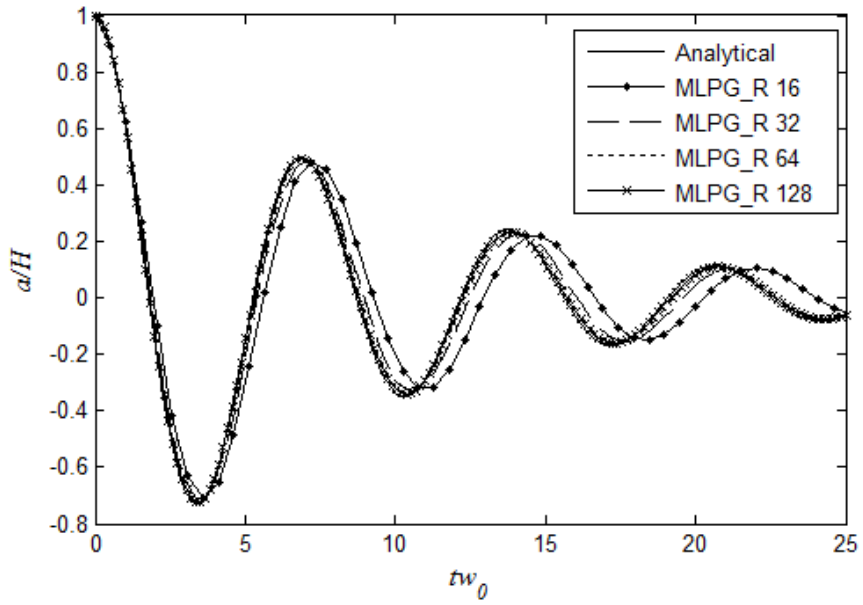
Figure 7.7: Equilibrium shape of the drop in (a) and corresponding pressure distribution in (b).

7.4.2 Numerical simulation of capillary wave

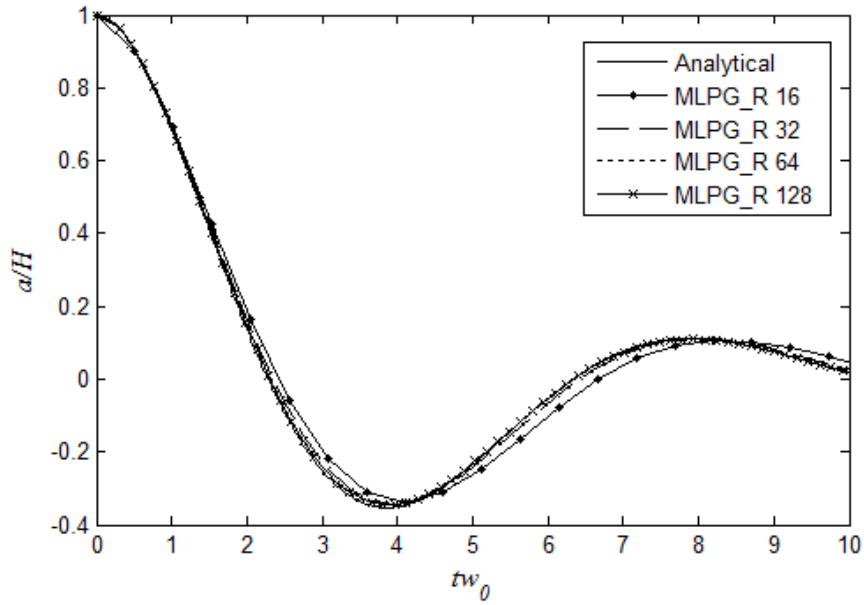
The capillary wave is surface tension driven flow and its small amplitude damped oscillation is a benchmark testing the accuracy of numerical schemes for viscous, surface tension driven two phase flows (Popinet 2009; Wang & Tong 2010). In this section, the accuracy of the newly proposed interface conditions, i.e., modified jumps of $\frac{\nabla p}{\rho}$ and jumps of p will be further tested. As the modified jump of $\frac{\nabla p}{\rho}$ is introduced for highly viscous fluids, the effects of $\frac{\nabla p}{\rho}$ jump modification for different viscosities will also be tested.

A computational domain of $H \times H$ is equally split into two sections filled by two fluids with identical kinematic viscosity. An initial sinusoidal perturbation is applied to the interface with the magnitude of $a_0 = 0.01H$ and the wave length of H. The solution of this initial value problem was found by Prosperetti (1981) restricted to linearized cases with identical kinematic viscosity. Due to infinite domains used in Prosperetti's theory, a periodic boundary is applied on the wave progressing direction. To achieve such a boundary, three columns of particles are added on the left and right sides of the domain. The right side added particles carry the information, i.e., density, viscosity, pressure and velocity, of three columns of particles on the far left of the computational domain and the left side added particles carry the information of three columns of particles on the far right of the computational domain. Following the settings in Wang & Tong (2010), the wave length is $\lambda = H = 1m$, the dynamic viscosity is $\mu_l = \mu_k = 1.0P_a \cdot s$, the density is $\rho_l = \rho_k = 100kg/m^3$ and the interfacial tension coefficient is $\sigma = 30N/m$, leading to the Ohnesorge number $Oh = \frac{\mu}{\sqrt{(\sigma\rho\lambda)}} = 1/\sqrt{3000}$. Convergence tests of different particle distance are first carried out. The wave amplitudes obtained by different particle numbers (16, 32, 64, 126) in a wave length are shown in Figure 7.8(a). Results of the test with the same parameters but a higher viscosity of $0.05\mathrm{m}^2/\mathrm{s}$ is shown in Figure 7.8(b), in which a stronger damping can be observed. For both cases, 64 particles in a wave length are sufficient to achieve a convergent results and this number will be applied to the following capillary wave cases unless otherwise stated. To observe the convergence rate, the error is calculated as $E = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(a_i - a_{exact,i}\right)^2}$ where N is the number of results in the time history, a_i is the amplitude from numerical simulation and $a_{exact,i}$ is from Prosperetti's analytical solution. For both viscosities of $0.01\,\mathrm{m}^2/\mathrm{s}$ and $0.05\,\mathrm{m}^2/\mathrm{s}$, the error E is calculated for different particle numbers which are 16, 32, 64 and 128 in a wave length.

Figure 7.9 plots the errors with respect to particle numbers, showing second order convergent rate for both viscosities.



(a)



(b)

Figure 7.8: Amplitudes time history with different particle numbers in a wave length. Kinematic viscosities are $0.01\mathrm{m}^2/\mathrm{s}$ in (a) and $0.05\mathrm{m}^2/\mathrm{s}$ in (b).

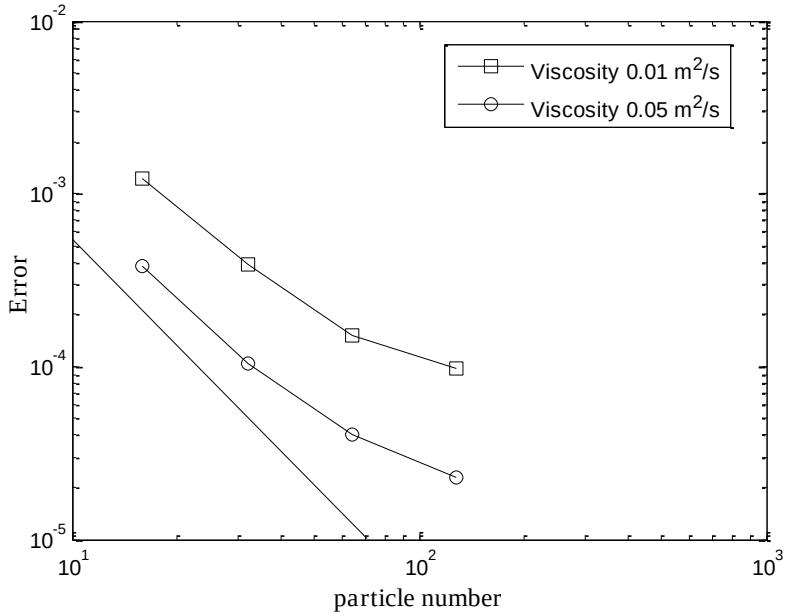
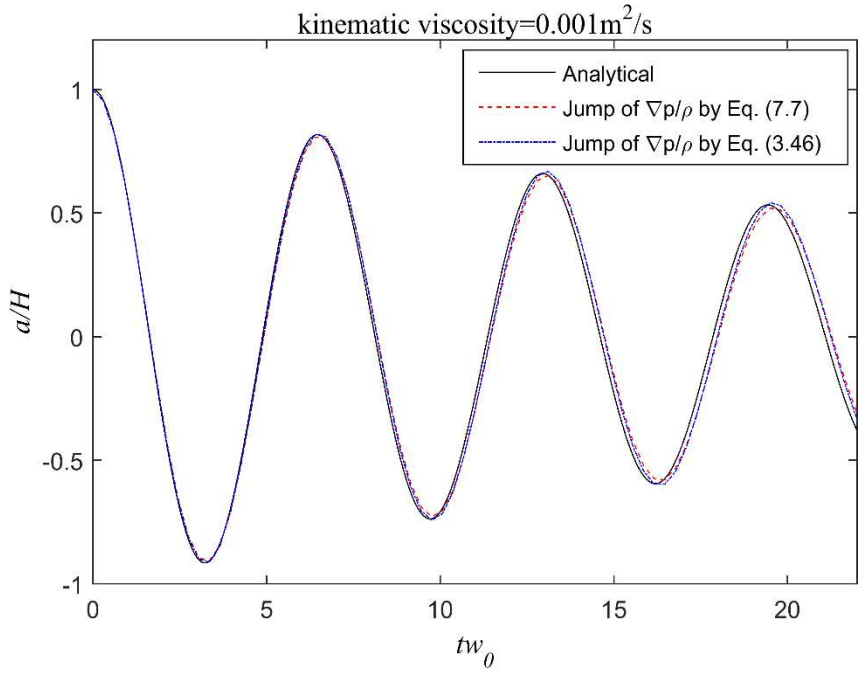
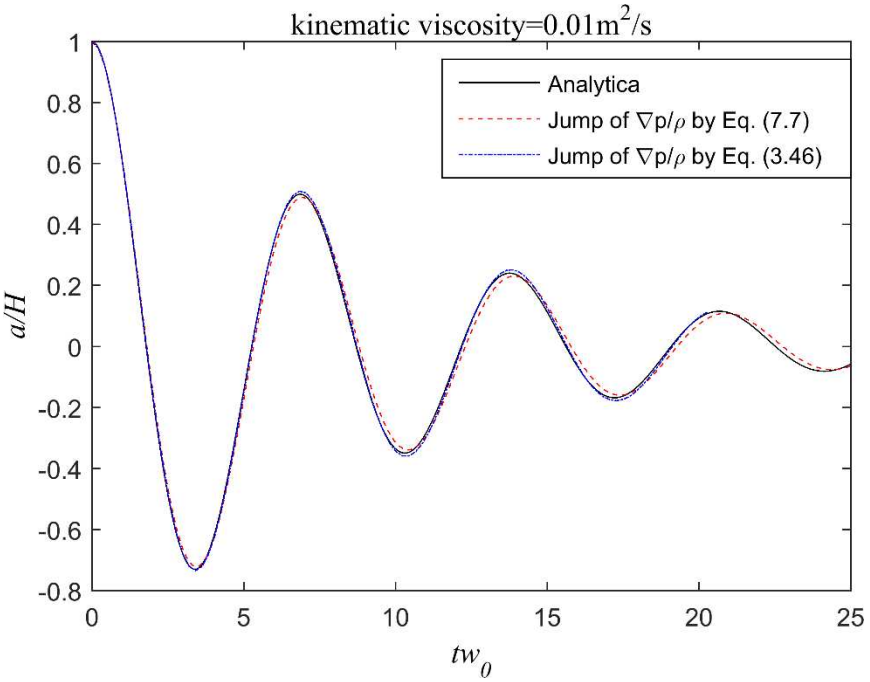


Figure 7.9: Errors for different particle numbers of 16, 32, 64 and 128 in a wave length for the viscosities of $0.01\mathrm{m}^2/\mathrm{s}$ and $0.05\mathrm{m}^2/\mathrm{s}$. The solid line indicates the second order convergence.

In Chapter 3, to ensure the continuity of $\frac{\partial p}{\rho \partial n}$ and also the normal velocity at the interface, the term of $\upsilon \nabla^2 \boldsymbol{u}$ for interface particles is enforced to be the average of two phases. Such treatment is applicable for less viscous fluids as stated in section 3.3 and this restriction will be further validated in this section. The viscosities of $0.001\mathrm{m}^2/\mathrm{s}$ and $0.01\mathrm{m}^2/\mathrm{s}$ are first tested with the jump of $\nabla p/\rho$ expressed by Eq. (3.46) and Eq. (7.7), giving the wave amplitudes shown in Figure 7.10(a) and (b). Results with both approaches almost coincide and agree well with the analytical solution. While as shown in Figure 7.10(c) and (d), with the higher viscosities of $0.05\mathrm{m}^2/\mathrm{s}$ and $0.075\mathrm{m}^2/\mathrm{s}$ the gaps in results obtained by the two approaches increase, showing that simulated amplitudes adopting the approach with the jump considering high viscous effects (i.e., Eq. (7.7)) in good agreement with the analytical solution while that obtained by the approach with the assumption of small viscous effects (i.e., Eq. (3.46)) deviates from the analytical solution, further as the viscosity increases. It is clear that the approach of Eq. (7.7) covers the range of both low and high viscosity whereas Eq. (3.46) only works well with low viscosity flows.

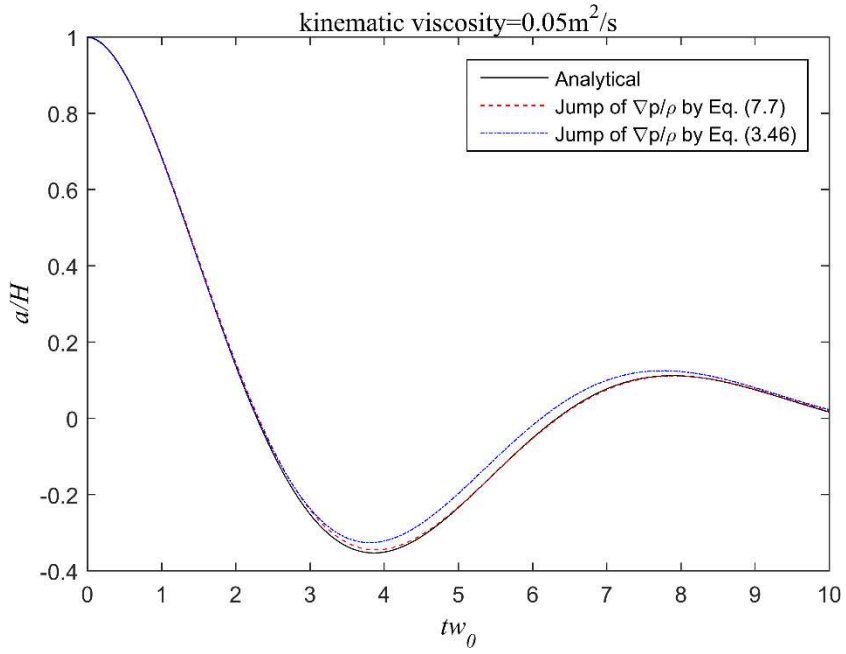


(a)

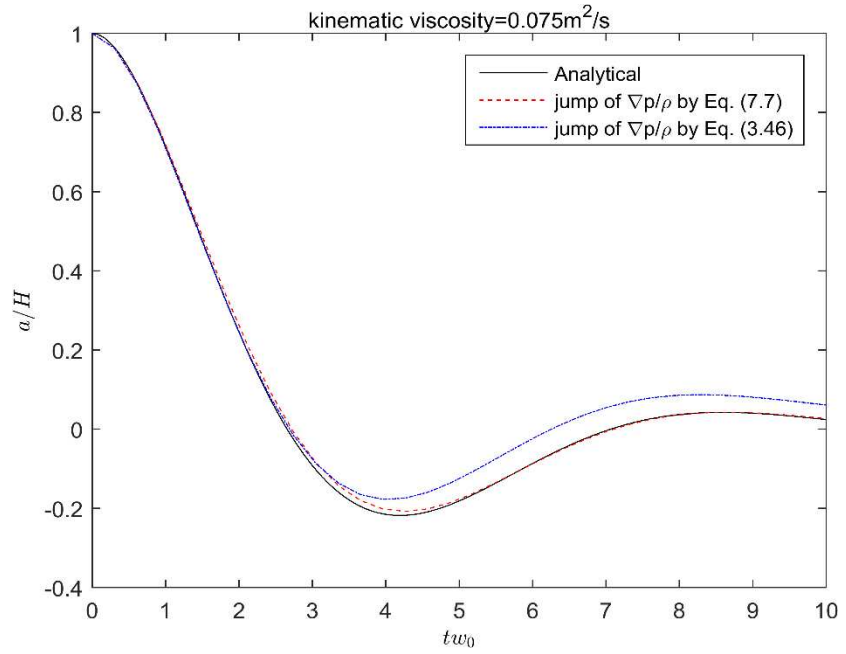


(b)

(To be continued on the next page)



(c)



(d)

Figure 7.10: Comparisons of amplitude time history between different expressions of the $\nabla p/\rho$ jump for viscosities of $0.001\text{m}^2/\text{s}$, $0.01\text{m}^2/\text{s}$, $0.05\text{m}^2/\text{s}$ and $0.075\text{m}^2/\text{s}$.

To quantitatively illustrate the deviations of numerical simulations from the analytical solutions, the relative error defined as $E_{r,i} = \left|a_i - a_{exact,i}\right|/\left|a_{exact,i}\right|$ is shown in Figure 7.11 for viscosities of $0.05\text{m}^2/\text{s}$ and $0.075\text{m}^2/\text{s}$. To avoid error enlargement due to the denominator approaching zero, the error is truncated when $\left|a_{exact,i}/a_0\right| \leq 0.04$. For either viscosity, the jump of $\nabla p/\rho$ by Eq. (3.46) leads to considerable errors. The error also goes up by increasing the viscosity, the maximum of which exceed 60% and 120%

for the values of $0.05\mathrm{m}^2/\mathrm{s}$ and $0.075\mathrm{m}^2/\mathrm{s}$ respectively. By modifying the jump, the error is significantly lowered and maintained below 5% for either viscosity. Comparisons of full interface profiles at two time instants for each viscosity, i.e. $tw_0 = 5.7$ and 6.4 for $0.05\mathrm{m}^2/\mathrm{s}$ as well as $tw_0 = 5.7$ and 7.6 for $0.075\mathrm{m}^2/\mathrm{s}$, are illustrated in Figure 7.12, further showing the significant improvement of the jump of $\nabla p/\rho$ by Eq. (7.7).

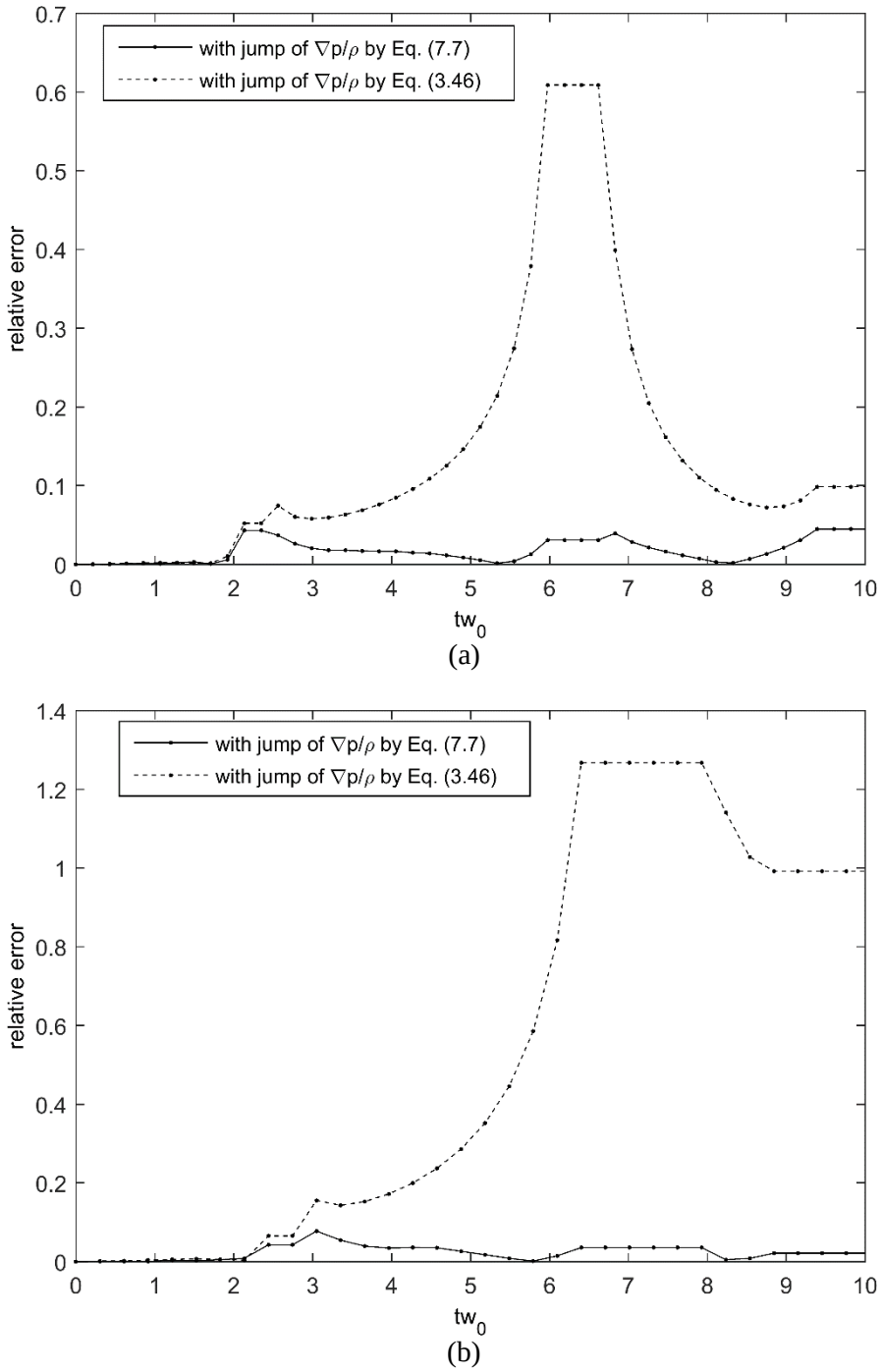
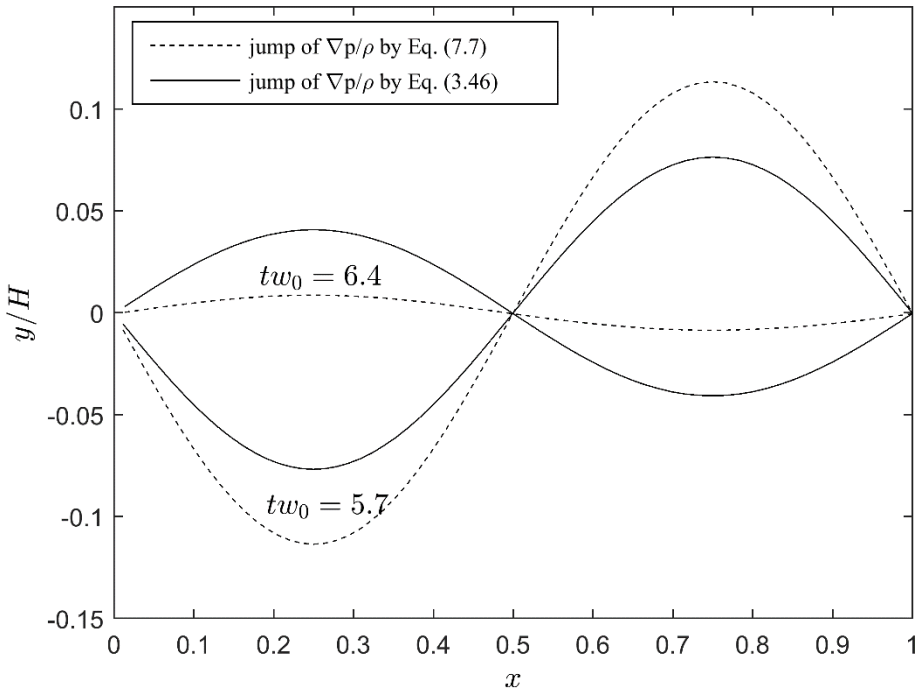
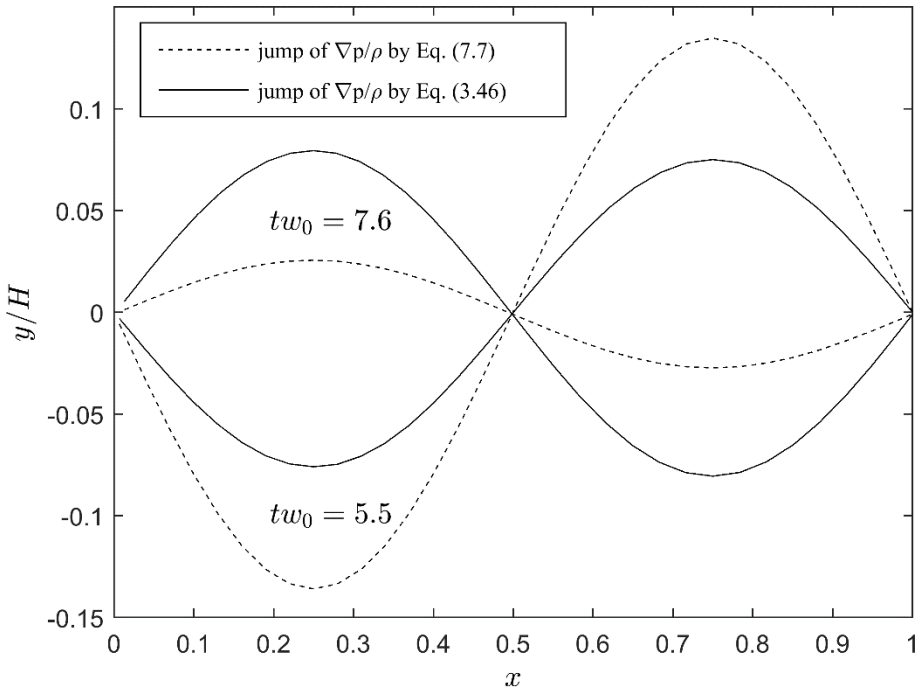


Figure 7.11: Relative errors for viscosities of $0.05\mathrm{m}^2/\mathrm{s}$ in (a) and $0.075\mathrm{m}^2/\mathrm{s}$ in (b). The error is truncated when $\left|a_{exact,i}/a_0\right| \leq 0.04$.



(a)



(b)

Figure 7.12: Interface profile comparisons at $tw_0 = 5.7$ *and* 6.4 for the viscosity of $0.05\text{m}^2/\text{s}$ in (a) and at $tw_0 = 5.5$ *and* 7.6 for the viscosity of $0.075\text{m}^2/\text{s}$ in (b).

7.4.3 Numerical simulation of bubble rising

In this section the proposed multiphase MLPG_R method with improved interface treatment is further tested and validated extensively against the published results through numerically simulating bubble rising in viscous fluids, the cases of which are strongly affected by high viscosity and surface tension. Such cases exam in the accuracy of surface tension and viscous terms estimation by the bubble shapes and rising velocities and were widely adopted for multiphase flow method validation (Zainali et al. 2013; Szewc et al. 2015).

Based on the definition in Hysing et al. (2009), a two dimensional computational domain with $H: B = 2: 1$ is adopted, imposing slip condition on left and right walls and non-slip boundary on top and bottom walls, see Figure 7.13. A bubble with the radius of $R = 0.25$ is initially allocated at $\left(\frac{B}{2}, 2R\right)$ where $B = 4R$. Non-dimensional densities assigned to fluids inside and outside the bubble are $\rho_k = 100$ and $\rho_l = 1000$ respectively. The viscosities are $\mu_k = 1$ and $\mu_l = 10$ and the interface tension coefficient is $\sigma = 24.5$. The flow is characterised by Reynolds number $Re = \frac{\rho_l U_g L}{\mu_l}$ and Bond number $Bo = \frac{\rho_l {U_g}^2 L}{\sigma}$, where the length scale is defined to be $L = 2R$ and the gravitational velocity is $U_g = \sqrt{2Rg}$. Re and Bo indicate viscous and surface tension effects against gravitational forces. In this case, $Re = 35$ and $Bo = 10$ indicating strong viscous and surface effects.

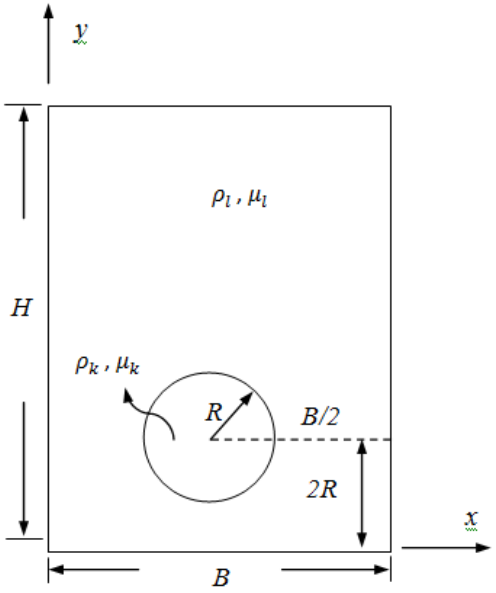


Figure 7.13: The setup of the bubble rising case

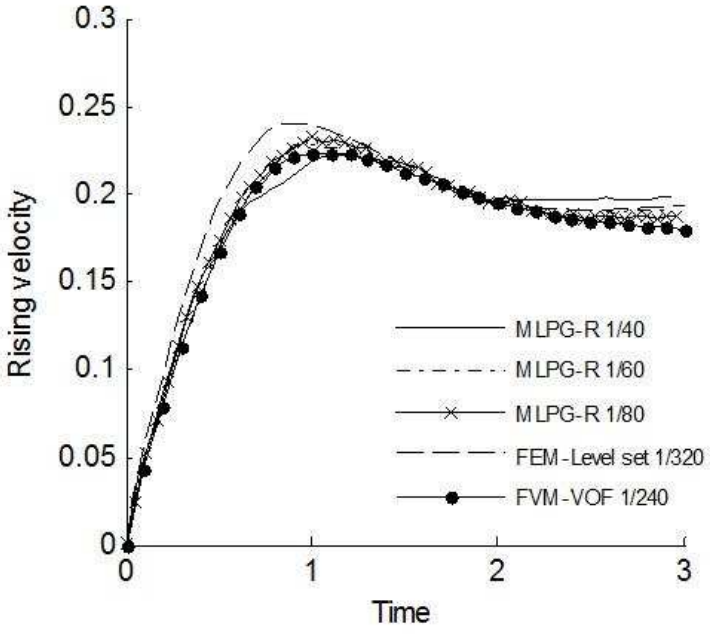
During bubble rising and shape translation, the bubble centroid position, $\boldsymbol{x}_c$, and velocity, $\boldsymbol{u}_c$, are usually used to track the bubble movement (Chen et al. 2005; Hysing et al. 2009) and are defined as

$$\boldsymbol{x}_c = \frac{\int_\Omega \boldsymbol{x} dx}{\int_\Omega 1 dx} \text{ and } \boldsymbol{u}_c = \frac{\int_\Omega \boldsymbol{u} dx}{\int_\Omega 1 dx},$$

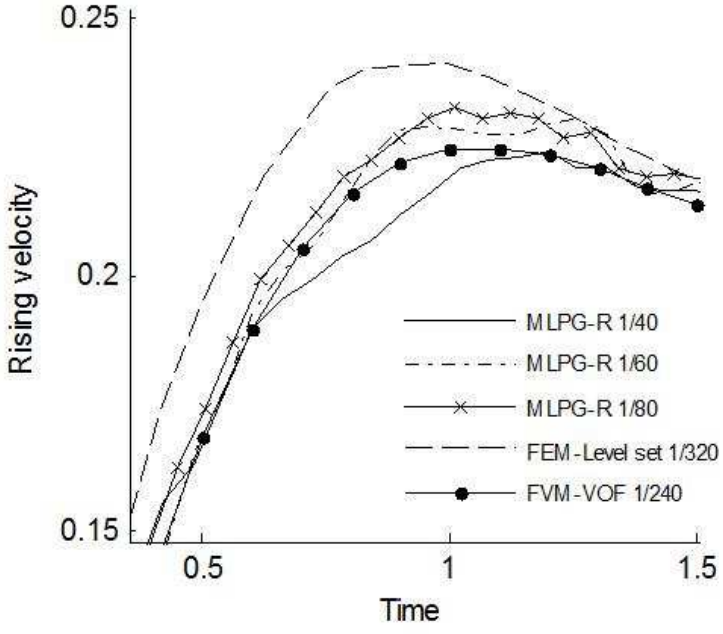
where Ω is the domain occupied by the bubble. The simulations of this case are conducted with three initial particle distances of $dl = 1/[80, 120, 160]$ to the nondimensional time $t = 3$. Besides the benchmark results from Hysing et al. (2009) in which centroid positions and rising velocities were obtained by FEM and the interface was captured by level set algorithm, those results from FVM using Volume of Fluid (VOF) capturing algorithm conducted in OpenFOAM which is an open source CFD solver are also presented.

Figure 7.14 compares the rising velocity for multiphase MLPG_R with particle distances of 1/40, 1/60 and 1/80, FEM of finest grids of 1/320 from Hysing et al. (2009) and FVM with refined and converged grids of 1/240 from OpenFOAM. The solutions from MLPG_R with different particle distances are close to each other but still have a noticeable gap between the results of Hysing et al. (2009). It can be observed from Figure 7.14(b) that the velocity converges when the distance is decreased to be 1/80 and this converged velocity locates between solutions of FEM and FVM. Compared to OpenFOAM's solution, the results obtained by multiphase MLPG_R are closer to the benchmark and it may be caused by the reducing of spurious currents which was previously discussed in the square-droplet case. Figure 7.15 illustrates the development of bubble centroid position by different particle distances of MLPG_R, FVM and FEM. Similar to the rising velocity, similar results are obtained by three MLPG_R resolutions which are also located between that of FVM and FEM.

According to numerical tests of MLPG_R, the bubble gets deformed as it rises. But it stays compact as one enclosed region throughout the simulation which is caused by the strong influence of the surface tension. The shape of the bubble finally becomes stable and the terminal shapes at t=3 obtained by different particle distances are shown in Figure 7.16. The shape also converges at $dl = 1/80$ for MLPG_R method and largely agrees with those from FEM capturing interface by Level set (Hysing et al. 2009) and FVM by VOF in OpenFOAM.

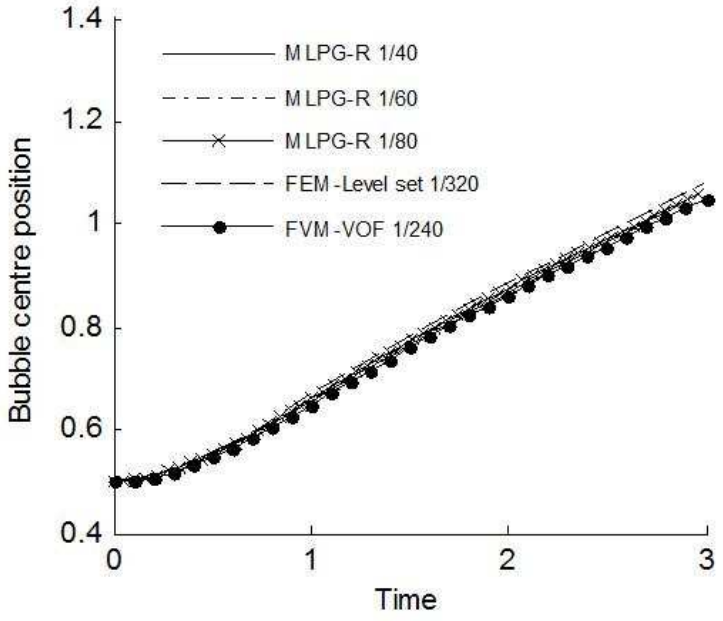


(a)

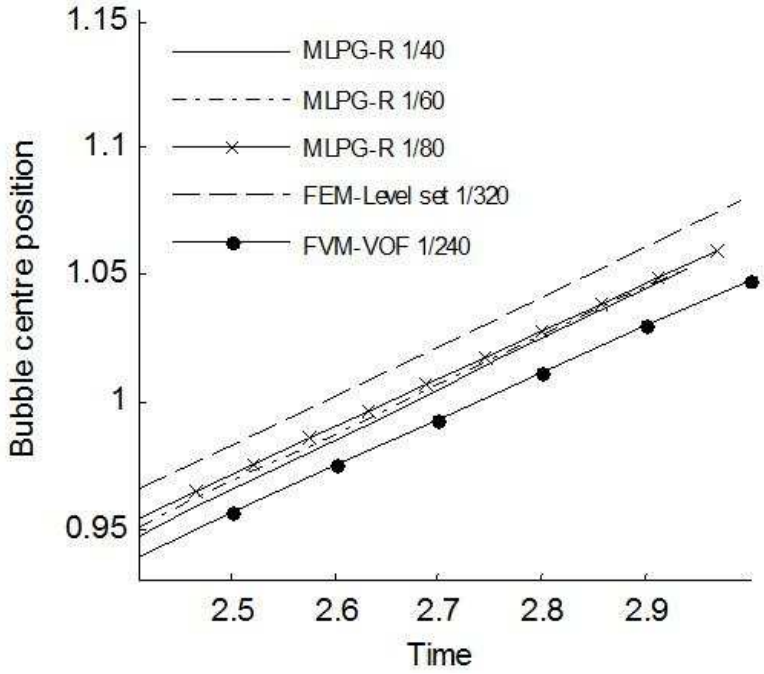


(b)

Figure 7.14: Rising velocities (a) from MLPG_R method with initial particle distance of 1/40, 1/60 and 1/80, FEM-Level set method (Hysing et al. 2009) and FVM-VOF (OpenFOAM) with very fine grids. (b) is details in maximum velocity region.



(a)



(b)

Figure 7.15: Centroid position of the bubble (a) from MLPG_R method with initial particle distance of 1/40, 1/60 and 1/80, FEM-Level set method (Hysing et al. 2009) and FVM-VOF (OpenFoam) with very fine grids. (b) is enlarged details.

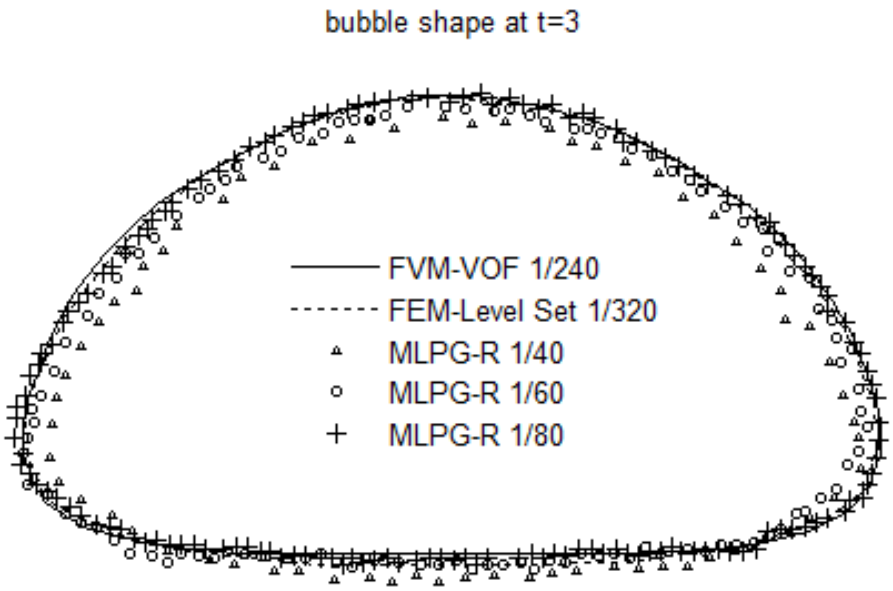


Figure 7.16: Bubble shape at t=3 from MLPG_R method with initial particle distance of 1/40, 1/60 and 1/80, FEM-Level set method (Hysing et al. 2009) and FVM-VOF (OpenFoam) with grids if 1/320 and 1/240 respectively.

Chapter 8 Conclusions

8.1 Numerical techniques for multiphase MLPG_R Method

This work has presented a two-phase flow model using the Meshless Local Petro-Galerkin method based on the Rankine source solution (MLPG_R) to simulate 2D flows of two immiscible fluids. A novel interface coupling method has been proposed to maintain discontinuity of the fluid properties and impose not only the dynamic interface condition of pressure continuity but also the kinematic condition of the ratio of pressure gradient to the density. By enforcing the above two interface conditions, the pressure at the interface is explicitly formulated using that of neighbouring particles within the support domain consisting of both phases. As the interface particles are identified at each time step representing a smooth interface consisting of one layer of interface particles, the newly formulated interface pressure can be exerted on the precise location. Those phase coupling treatments enable the model to cover a wide range of density ratios from 1.01 to 1000 under either non-breaking or breaking wave situations.

To further extend the model to be applicable for surface tension and viscous effect dominated cases, the implementation of the interface conditions has also been improved to satisfy the stress balance including interface tension in the normal direction of the interface and the velocity continuity at the interface. Based on the new interface conditions, a pressure jump is added and the ratio of pressure gradient to the density is modified, leading to additional terms incorporated into the original interface pressure formulation. As interface tension is involved in the simulation and its accuracy relies highly on the curvature estimation, the curvature is analytically worked out from the local

reconstructed interface curve which is based on the precisely identified interface particles. Tests on different curve types are carried out and second order convergence of the particle distance is achieved when the second order moving least square (MLS) approach is adopted to reconstruct the local interface curve.

Another crucial issue addressed in this study is the interface particle identification. With precisely identified interface particles, the interface conditions can be properly implemented and the curvature of interface can also be accurately determined. The new interface particle identification method developed in this work is based on the absolute density gradient, having a numerical value very close to 0 for inner particles and rapidly increasing when approaching the interface or isolated particles. The performance of the method has been tested for different configurations including those with different levels of randomness of particle distributions. It is shown that the accuracy of the method is independent of the density ratio of two phases and of the average distance between particles. In all the cases tested, the method can correctly pick up almost all the interface and isolated particles as long as the support domain is larger than twice the average distance of particles.

In this study, the two approaches have also been proposed to solve the algebraic pressure equation. One is that both phases are solved in one set of equations (Integrated-1) and the other splits the equations into two sets which are then solved separately within each phase (Coupled-2). In Coupled-2 approach, as the interface pressure is based on the pressure of the surrounding particles of the last time step, validations for the results obtained by Coupled-2 method with or without iterations between two sets of equations are carried out and it is found that iteration is not necessary when the time step is sufficiently small. For the comparison between two approaches by numerical tests, it came to the conclusions (1) the Coupled-2 approach works well for a very high density ratio (even as high as 1000) while the Integrated-1 approach is only limited to flows with small density ratio (typically less than 1.3) and (2) where both approaches work equally well, the Coupled-2 approach is computationally more efficient.

In summary, the numerical techniques proposed in this study achieves not only the extension of MLPG_R method to two-phase flow modelling but also bring robustness compared to existing two-phase flow models as listed below:

- The model maintains sharp discontinuity of fluid properties across the interface and the interface with zero thickness.

- Complete and accurate interface conditions are explicitly implemented at the sharp interface with a newly derived pressure expression.
- Coupled-2 approach effectively solves pressure algebraic equation
- The model achieves simulations for the flows with high density ratio (to 1000), being violent and breaking and with high interface tension and viscous effects without any artificial compensations.

8.2 Model validation and application

Model validations and applications have been conducted in two groups of cases corresponding to two interface implementation stages by incorporating the newly proposed interface particle identification technique and choosing the Coupled-2 approach to solve pressure equations. The first group is carried out by the interface conditions of continuous pressure and the continuous ratio of the normal pressure derivative to the density considering negligible surface tension and small viscosity. A jump on the ratio of the pressure gradient to the density is maintained for the consideration of disordered particle distribution. The model was first validated by the gravity current case with density ratio ranging from 1.01 to 1.3. It was found that the progress the dense phase front is validated by experimental data showing a constant front velocity and the predictions agreed well with the experiment. Being compared with experimental shadow graphs of flow configurations, the current at three time instants have been well reproduced and the pressure has been found continuous at the interface and smoothly distributed without any noise. Simulations of natural sloshing with small amplitude in a tank fully filled by two-layer liquid have also been carried out to validate the model with analytical solutions involving wave elevations and pressure. Good agreements of interface elevation time history, pressure distribution across the interface for a number of time instants have been reached. With density ratio ranging from 10 to 1000 and dense fluid filling ratio of 0.5 and 0.3, all the tests conducted on wave elevations have achieved second order convergence with initial particle distance. Further validations on the natural frequencies for the filling ratio from 0.1 to 0.9 almost coincide with the analytical solutions. The last case in this group has been concerned with violent air-water sloshing with density ratio of 1000 in which a jet of water impinges on the wall and wave breaking occurs, leaving isolated water particles splashing out and air particles penetrating into the water. The wave profiles obtained by the simulation broadly agree with the test results as revealed by experiment photos. The predicted pressure time history on the wall is smooth with

negligible numerical induced oscillations and shows a good agreement when compared with experimental data.

The second group has been carried out by the interface condition with a jump on the pressure and the jump on the ratio of the pressure gradient to the density is modified, including the effects of interface tension and large viscosity. The model has been validated by the case of square-droplet deformation in which the initial square is found gradually deform to become a circle due to the presence of interface tension and finally reach an equilibrium state. As the interface is sharply represented and a sharp pressure jump is implemented at the interface, the pressure achieves two constants inside and outside the circle by completing the jump on the interface particle, a result that agrees well with Laplace's Law. Benefitting from the precisely implemented interface conditions, the spurious current phenomena has been significantly relieved when compared with volume of fluid (VOF) interface treatment illustrated by the reduced maximum residual velocity at the equilibrium state. Another validation case is the capillary wave simulation, amplitude damping with the analytical solution has been compared and the effect of the jump modification on the ratio of the pressure gradient over the density has been discussed. The damping of the interface amplitude for different viscosities is obtained by either interface condition with two different jumps on $\frac{\nabla p}{\rho}$ and compared with the analytical solutions. It comes to the conclusion that as the viscosity increases, the results obtained by the condition with the jump of $\frac{\nabla p}{\rho}$ considering small viscosity deviate further from the analytical solutions while that obtained by the condition with the jump of $\frac{\nabla p}{\rho}$ considering large viscosity maintain good agreement with the analytical solutions. The last case is bubble rising that has been validated extensively against the published results through numerically simulating bubble rising in viscous fluids. The final bubble shape predicted by the present MLPG_R method is close to the ones obtained by finite element method (FEM) with level set and finite volume method (FVM) with VOF. As for the rising velocity and centroid position progress, results of multiphase MLPG_R method locate between those obtained by these two methods.

According to case studies, the model has been validated to be effective and have favourable features as

- Continuous pressure across the interface for the cases without interface tension;
- Sharp pressure jump at the interface presenting the interface tension if it is considered;

- Pressure away from the interface is smoothly varied without notable numerically caused oscillations;
- Second order convergence is achieved;
- Applications cover the density ratio from 1 to 1000 including violent flows with breakings.

8.3 Recommendation for future works

Although the current multiphase MLPG_R method has been validated against a large number of cases, improvements in many aspects of the method are recommended here to enable the method to become more applicable in situations of reality, to stretch it for more diversity of applications by considering other complex behaviour of the multiphase flow and to speed up the simulation.

As only a 2D model is developed in this study, the extension to a 3D model is essential to simulate a wider range of cases in reality such as a spherical bubble rising in the water column and the sloshing occurring in a ship carrying oil or at LNG tank which cannot be simplified as a 2D tank. For the simulation of sediment (sand or mud) transportation in water (Shao & Lo 2003; Shakibaeinia & Jin 2012a), even though fluid-sediment mixtures can be treated as continuous fluid, the non-Newtonian property of the flow has to be considered. Turbulence commonly exists in violent breaking waves, pipe flows, jets and wakes, etc. which have been modelled as single phase flow (Gotoh & Sakai 2006; Dalrymple & Rogers 2006). When considered as multiphase flows, the simulation of turbulence becomes challenging. The compressibility effects of the air related to breaking waves and in other more general situations have to be considered. From the point of view of computational efficiency, parallel computation or even graphic processing unit (GPU) utilization is necessary for large domain simulation, especially for 3D cases. An alternative way to speed up the simulation in ocean water wave interaction is to develop an integrated model using multiphase MLPG_R to deal with local wave breaking, air entrapment and viscous effect while a rough but more efficient model (i.e. based on potential theory) to deal with outer domain.

References

Adalsteinsson, D. & Sethian, J.A., 1995. A Fast Level Set Method for Propagating Interfaces. Journal of Computational Physics, 118(2), pp.269–277.

Adami, S., Hu, X.Y. & Adams, N.A., 2010. A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation. Journal of Computational Physics, 229(13), pp.5011–5021.

Anderson, D.M., McFadden, G.B. & Wheeler, a. a., 1998. Diffuse-Interface Methods in Fluid Mechanics. Annual Review of Fluid Mechanics.

Antuono, M., Colagrossi, A. & Marrone, S., 2012. Numerical diffusive terms in weakly-compressible SPH schemes. Computer Physics Communications, 183(12), pp.2570–2580.

Ataie-Ashtiani, B. & Farhadi, L., 2006. A stable moving-particle semi-implicit method for free surface flows. Fluid Dynamics Research, 38(4), pp.241–256.

Atluri, S.N., Liu, H.T. & Han, Z.D., 2006. Meshless Local Petrov-Galerkin (MLPG) mixed finite difference method for solid mechanics. CMES - Computer Modeling in Engineering and Sciences, 15(1), pp.1–16.

Atluri, S.N. & Shen, S., 2002. The Meshless Local Petrov-Galerkin (MLPG) method: A simple & less-costly alternative to the finite element and boundary element methods. CMES - Computer Modeling in Engineering and Sciences, 3(1), pp.11–51.

Atluri, S.N. & Zhu, T., 1998. A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. Computational Mechanics, 22(2), pp.117–127.

Atluri, S.N. & Zhu, T., 2000. New concepts in meshless methods. International Journal for Numerical Methods in Engineering, 47(1-3), pp.537–556.

Aulisa, E., Manservisi, S., Scardovelli, R. & Zaleski, S., 2007. Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry. Journal of Computational Physics, 225(2), pp.2301–2319.

Ausas, R.F., Dari, E.A. & Buscaglia, G.C., 2011. Ageometric mass-preserving redistancing scheme for the level set function Roberto. International Journal for Numerical Methods in Fluids, 65, pp.989–1010.

Batra, R.C. & Ching, H.K., 2002. Analysis of elastodynamic deformations near a crack/notch tip by the meshless local Petrov-Galerkin (MLPG) method. Computational Mechanics, 3(6), pp.717–730.

Bhaga, D. & Weber, M.E., 1981. Bubbles in viscous liquids: shapes, wakes and velocities. Journal of Fluid Mechanics, 105, pp.61–85.

Bothe, D., Kröger, M. & Warnecke, H.J., 2011. A VOF-based conservative method for the simulation of reactive mass transfer from rising bubbles. Fluid Dynamics and Materials Processing, 7(3), pp.303–316.

Brackbill, J.U., Kothe, D.B. & Zemach, C., 1992. A continuum method for modeling surface tension. Journal of Computational Physics, 100, pp.335–354.

Buwa, V. V., Deo, D.S. & Ranade, V. V., 2006. Eulerian–Lagrangian simulations of unsteady gas–liquid flows in bubble columns. International Journal of Multiphase Flow, 32(7), pp.864–885.

Chen, T., Minev, P.D. & Nandakumar, K., 2005. A projection scheme for incompressible multiphase flow using adaptive Eulerian grid: 3D validation. International Journal for Numerical Methods in Fluids, 48(4), pp.455–466.

Chen, X., Xi, G. & Sun, Z.G., 2014. Improving stability of MPS method by a computational scheme based on conceptual particles. Computer Methods in Applied Mechanics and Engineering, 278, pp.254–271.

Chen, Z., Zong, Z., Liu, M.B., Zou, L., Li, H.T. & Shu, C., 2015. An SPH model for multiphase flows with complex interfaces and large density differences. Journal of Computational Physics, 283, pp.169–188.

Cleary, P.W., Ha, J., Mooney, J, & Ahuja, V., 1998. Effect of heat transfer and solidification on high pressure die casting. In 13th Australasian Fluid Mechanics Conference. Melbourne, pp. 679–682.

Clift, R., Grace, J.R. & Weber, M.E., 1978. Bubbles, Drops and Particles, New York: Academic Press.

Colagrossi, A., 2005. A Meshless Lagrangian Method for Free – Surface and Interface Flows with Fragmentation. , p.233.

Colagrossi, A. & Landrini, M., 2003. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. Journal of Computational Physics, 191(2), pp.448–475.

Cummins, S.J., Francois, M.M. & Kothe, D.B., 2005. Estimating curvature from volume fractions. Computers and Structures, 83(6-7), pp.425–434.

Dalrymple, R.A. & Rogers, B.D., 2006. Numerical modeling of water waves with the SPH method. Coastal Engineering, 53(2-3), pp.141–147.

Das, A.K. & Das, P.K., 2011. Incorporation of diffuse interface in smoothed particle hydrodynamics: Implementation of the scheme and case studies. International Journal for Numerical Methods in Fluids, 67, pp.671–699.

Das, D.B. & Mirzaei, M., 2012. Dynamic Effects in Capillary Pressure Relationships for Two-Phase Flow in Porous Media: Experiments and Numerical Analyses. AIChE Journal, 58(12), pp.3891–3903.

Delnoij, E., Lammers, F.A., Kuipers, J.A.M. & van Swaaij, W.P.M., 1997. Dynamic simulation of dispersed gas-liquid two-phase flow using a discrete bubble model. Chemical Engineering Science, 52(9), pp.1429–1458.

Desjardins, O., Moureau, V. & Pitsch, H., 2008. An accurate conservative level set/ghost

fluid method for simulating turbulent atomization. Journal of Computational Physics, 227(18), pp.8395–8416.

Dieter-Kissling, K., Karbaschi, M., Marschall, H., Javadi, A., Miller, R & Bothe, D., 2014. On the applicability of Drop Profile Analysis Tensiometry at high flow rates using an interface tracking method. Colloids and Surfaces A: Physicochemical and Engineering Aspects, 441, pp.837–845.

Diwakar, S. V., Das, S.K. & Sundararajan, T., 2009. A Quadratic Spline based Interface (QUASI) reconstruction algorithm for accurate tracking of two-phase flows. Journal of Computational Physics, 228(24), pp.9107–9130.

Drazin, P.G. & Reid, W.H., 1981. Hydrodynamic Stability, Cambridge: Cambridge University Press.

Druzhinin, O.A. & Elghobashi, S., 1998. Direct numerical simulations of bubble-laden turbulent flows using the two-fluid formulation. Physics of Fluids, 10(3), p.685.

Faltinsen, O.M. & Timokha, A.N., 2009. Sloshing, New York: cambridge university press.

Fatehi, R. & Manzari, M.T., 2011. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. Computers and Mathematics with Applications, 61(2), pp.482–498.

Fedkiw, R.P., Aslam, T., Merriman, B. & Osher, S., 1999. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). Journal of Computational Physics, 152(2), pp.457–492.

Flekkoy, E.G., Coveney, P.V. & De Fabritiis, G., 2000. Foundations of dissipative particle dynamics. Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics, 62(2), p.2140.

Gauvin, W.H., Katta, S. & Knelman, F.H., 1975. Drop trajectory predictions and their importance in the design of spray dryers. International Journal of Multiphase Flow, 1(6), pp.793–816.

Ginzburg, I. & Wittum, G., 2001. Two-Phase Flows on Interface Refined Grids Modeled with VOF, Staggered Finite Volumes, and Spline Interpolants. Journal of Computational Physics, 166(2), pp.302–335.

Gotoh, H., Khayyer, A., Arikawa, T. & Shimosako, K., 2014. On enhancement of Incompressible SPH method for simulation of violent sloshing flows. Applied Ocean Research, 46, pp.104–115.

Gotoh, H. & Fredsøe, J., 2000. Lagrangian Two-Phase Flow Model of the Settling Behavior of Fine Sediment Dumped into Water. Coastal Engineering, (1997), pp.3906–3919.

Gotoh, H. & Sakai, T., 2006. Key issues in the particle method for computation of wave breaking. Coastal Engineering, 53(2-3), pp.171–179.

Gotoh, H. & Sakai, T., 1999. Lagrangian Simulation of Breaking Waves Using Particle Method. Coastal Engineering Journal, 3(4), pp.303–326.

Gotoh, H., Shibahara, T. & Sakai, T., 2001. Sub-particle-scale turbulence model for the MPS method - Lagrangian flow model for hydraulic engineering. Advanced Methods for Computational Fluid Dynamics, 9-4, pp.339 – 347.

Grenier, N., Antuono, M., Colagrossi, A., Le Touzé, D. & Alessandrini, B., 2009. An Hamiltonian interface SPH formulation for multi-fluid and free surface flows. Journal of Computational Physics, 228(22), pp.8380–8393.

Grobelbauer, H.P., Fannelop, T.K. & Britter, R.E., 1993. The prograda tion of intrusion fronts of high density ratios. Journal of Fluid Mechanics, 250, pp.669–687. Available at: -.

Gui, Q., Shao, S.D. & Dong, P., 2014. Wave Impact Simulations by an Improved Incompressible SPH Model. Journal of Waterway, Port, Coastal, and Ocean Engineering, 140(3).

Guo, Z. & Lin, P., 2015. A thermodynamically consistent phase-field model for two-phase flows with thermocapillary effects. Journal of Fluid Mechanics, 766, pp.226–271.

Haj-Hariri, H., Shi, Q. & Borhan, A., 1997. Thermocapillary motion of deformable drops at finite Reynolds and Marangoni numbers. Physics of Fluids, 9(4), pp.845–855.

Han, Z.D., Liu, H. T., Rajendran, A.M. & Atluri, S.N., 2006. The Applications of Meshless Local Petrov-Galerkin (MLPG) Approaches in High-Speed Impact , Penetration and Perforation Problems. CMES - Computer Modeling in Engineering and Sciences, 14(2), pp.119–128.

Han, Z.D. & Atluri, S.N., 2004a. A Meshless Local Petrov-Galerkin (MLPG) approach for 3-dimensional elasto-dynamics. Computers, Materials and Continua, 1(2), pp.129–140.

Han, Z.D. & Atluri, S.N., 2004b. Meshless local Petrov-Galerkin (MLPG) approaches for solving 3D problems in elasto-statics. CMES - Computer Modeling in Engineering and Sciences, 6(2), pp.169–188.

He, Q. & Kasagi, N., 2007. Phase-Field simulation of small capillary- number two-phase flow in a microtube. Fluid Dynamics Research, 40(7-8), pp.497–509.

Hirt, C.W. & Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics, 39(1), pp.201–225.

Hoang, D.A., van Steijn, V., Portela, L.M., Kreutzer, M.T. & Kleijn, C.R., 2013. Benchmark numerical simulations of segmented two-phase flows in microchannels using the Volume of Fluid method. Computers and Fluids, 86, pp.28–36.

Holmes, D.W., Williams, J.R. & Tilke, P., 2011. Smooth particle hydrodynamics simulations of low Reynolds number flows through porous media. International Journal for Numerical and Analytical Methods in Geomechanics, 35(4), pp.419–437.

Hu, X.Y. & Adams, N.A., 2006. A multi-phase SPH method for macroscopic and

mesoscopic flows. Journal of Computational Physics, 213(2), pp.844–861.

Hu, X.Y. & Adams, N.A., 2009. A constant-density approach for incompressible multi-phase SPH. Journal of Computational Physics, 228(6), pp.2082–2091.

Hu, X.Y. & Adams, N.A., 2007. An incompressible multi-phase SPH method. Journal of Computational Physics, 227(1), pp.264–278.

Hwang, S.C., Khayyer, A., Gotoh, H. & Park, J.C., 2014. Development of a fully Lagrangian MPS-based coupled method for simulation of fluid–structure interaction problems. Journal of Fluids and Structures, 50, pp.497–511.

Hysing, S., 2006. A new implicit surface tension implementation for interfacial flows. International Journal for Numerical Methods in Fluids, 51(6), pp.659–672.

Hysing, S., Turek, S., Kuzmin, D., Parolini, N., Burman, E. & Ganesan, S., 2009. Quantitative benchmark computations of two-dimensional bubble dynamics. International Journal for Numerical Methods in Fluids, (60), pp.1259–1288.

Kang, M., Fedkiw, R.P. & Liu, X.D., 2000. A boundary condition capturing method for multiphase incompressible flow. Journal of Scientific Computing, 15(3), pp.323–360.

Keller, J.J. & Chyou, Y.P., 1991. On the hydraulic lock-exchange problem. Journal of Mathematics and Physics, 42(November), pp.874–910.

Khayyer, A. & Gotoh, H., 2012. A 3D higher order Laplacian model for enhancement and stabilization of pressure calculation in 3D MPS-based simulations. Applied Ocean Research, 37, pp.120–126.

Khayyer, A. & Gotoh, H., 2010. A higher order Laplacian model for enhancement and stabilization of pressure calculation in 3D MPS-based simulations. Applied Ocean Research, 37(1), pp.120–126.

Khayyer, A. & Gotoh, H., 2008. DEVELOPMENT OF CMPS METHOD FOR ACCURATE WATER-SURFACE TRACKING IN BREAKING WAVES. Coastal Engineering Journal, 50(02), pp.179–207.

Khayyer, A. & Gotoh, H., 2013. Enhancement of performance and stability of MPS mesh-free particle method for multiphase flows characterized by high density ratios. Journal of Computational Physics, 242, pp.211–233.

Khayyer, A. & Gotoh, H., 2009. Modified Moving Particle Semi-implicit methods for the prediction of 2D wave impact pressure. Coastal Engineering, 56(4), pp.419–440.

Kishev, Z.R., Hu, C. & Kashiwagi, M., 2006. Numerical simulation of violent sloshing by a CIP-based method. Journal of Marine Science and Technology, 11(2), pp.111–122.

Komatina, D. & Jovanovic, M., 1997. Experimental study of steady and unsteady free surface flows with water-clay mixtures. Journal of Hydraulic Research, 35(5), pp.579–590.

Kondo, M. & Koshizuka, S., 2011. Improvement of stability in moving particle semi-

implicit method. International Journal for Numerical Methods in Fluids, (October 2007), pp.638–654.

Koshizuka, S., Nobe, A. & Oka, Y., 1998. Numerical analysis of breaking waves using the moving particle semi-implicit method. International Journal for Numerical Methods in Fluids, 26(7), pp.751–769.

Koshizuka, S. & Oka, Y., 1996. Moving-particle semi-implicit method for fragmentation of incompressible fluid. Nuclear science and engineering, 123(3), pp.421–434.

Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S. & Zanetti, G., 1994. Modelling Merging and Fragmentation in Multiphase Flows with SURFER. Journal of Computational Physics, 113(1), pp.134–147.

Landau, L.D. & Lifshitz, E.M., 1987. Fluid Mechanics 2nd ed., Oxford: cambridge university press.

Lee, E.S., Moulinec, C., Xu, R., Violeau, D., Laurence, D. & Stansby, P., 2008. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. Journal of Computational Physics, 227(18), pp.8417–8436.

Lin, H. & Atluri, S.N., 2000. Meshless Local Petrov-Galerkin (MLPG) Method for Convection-Diffusion Problems. Cmes, 1(2), pp.45–60.

Lin, H. & Atluri, S.N., 2001. The Meshless Local Petrov-Galerkin (MLPG) method for solving incompressible Navier-Stokes equations. CMES - Computer Modeling in Engineering and Sciences, 2(2), pp.117–142.

Lind, S.J., Xu, R., Stansby, P. K. & Rogers, B. D., 2012. Incompressible smoothed particle hydrodynamics for free: surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. Journal of Computational Physics, 231(4), pp.1499–1523.

Liu, C. & Shen, J., 2003. A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method. Physica D: Nonlinear Phenomena, 179(3-4), pp.211–228.

Liu, G.R. & Liu, M.B., 2003. Smoothed particle hydrodynamics: a meshfree particle method, World Scientific.

Liu, X., Xu, H., Shao, S.D. & Lin, P., 2013. An improved incompressible SPH model for simulation of wave-structure interaction. Computers and Fluids, 71, pp.113–123.

López, J., Hernández, J., Gómez, P. & Faura, F., 2004. A volume of fluid method based on multidimensional advection and spline interface reconstruction. Journal of Computational Physics, 195(2), pp.718–742.

Lowe, R.J., Rottman, J.W. & Linden, P.F., 2005. The non-Boussinesq lock-exchange problem. Part 1. Theory and experiments. Journal of Fluid Mechanics, 537, pp.101–124.

Lowengrub, J. & Truskinovsky, L., 1998. Quasi-incompressible Cahn-Hilliard fluids and

topological transitions. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 454(1978), pp.2617–2654.

Lucy, L.B., 1977. A numerical approach to the testing of fission hypothesis. Astronomical Journal, 82, pp.1013–1024.

Ma, C. & Bothe, D., 2011. Direct numerical simulation of thermocapillary flow based on the Volume of Fluid method. International Journal of Multiphase Flow, 37(9), pp.1045–1058.

Ma, Q.W., 2008. A New Meshless Interpolation Scheme for MLPG_R Method. CMES - Computer Modeling in Engineering and Sciences, 23(2), pp.75–89.

Ma, Q.W., 2005a. Meshless local Petrov-Galerkin method for two-dimensional nonlinear water wave problems. Journal of Computational Physics, 205(2), pp.611–625.

Ma, Q.W., 2005b. MLPG Method Based on Rankine Source Solution for Simulating Nonlinear Water Waves. CMES - Computer Modeling in Engineering and Sciences, 9(2), pp.193–209.

Ma, Q.W., 2007. Numerical generation of freak waves using MLPG_R and QALE-FEM methods. CMES - Computer Modeling in Engineering and Sciences, 18(3), pp.223–234.

Ma, Q.W. & Zhou, J.T., 2009. MLPG_R Method for Numerical Simulation of 2D BreakingWaves. CMES - Computer Modeling in Engineering and Sciences, 43(3), pp.277–303.

Min, C., 2010. On reinitializing level set functions. Journal of Computational Physics, 229(8), pp.2764–2772.

Monaghan, J.J., Cas, R.A.F., Kos, A.M. & Hallworth, M., 1999. Gravity currents descending a ramp in a stratified tank. Journal of Fluid Mechanics, 379, pp.39–69.

Monaghan, J.J., 1994. Simulating Free Surface Flows with SPH. Journal of Computational Physics, 110(2), pp.399–406.

Monaghan, J.J., 2000. Simulating surface tension with smoothed particle hydrodynamics. International Journal for Numerical Methods in Fluids, 33(3), pp.333–353.

Monaghan, J.J., 2005. Smoothed Particle Hydrodynamics. Reports on progress in physics, 68(8), p.1703.

Monaghan, J.J. & Kocharyan, A., 1995. SPH simulation of multi-phase flow. Computer Physics Communications, 87(1-2), pp.225–235.

Monaghan, J.J. & Rafiee, A., 2013. A simple SPH algorithm for multi-fluid flow with high density ratios. International Journal for Numerical Methods in Fluids, 71, pp.537–561.

Morris, J.P. & Monaghan, J.J., 1997. A Switch to Reduce SPH Viscosity. Journal of Computational Physics, 136(1), pp.41–50.

Morris, J.P., Zhu, Y. & Fox, P.J., 1999. Parallel simulations of pore-scale flow through

porous media. Computers and Geotechnics, 25(4), pp.227–246.

Muzaferija, S. & Peri´c, M., 1999. Computation of flows using iterface-tracking and interface-capturingm methods. In O. Mahrenholtz & Markiewicz.M, eds. Nonlinear Water Wave Interaction. Southampton: WIT Press, pp. 59–100.

Muzaferija, S. & Peri´c, M., 1997. Computation of Free-Surface Flows Using the Finite-Volume Method and Moving Grids. Numerical Heat Transfer, Part B: Fundamentals, 32(4), pp.369–384.

Muzaferija, S. & Peri´c, M., 1997. Computation of Free-Surface Flows Using the Finite-Volume Method and Moving Grids. Numerical Heat Transfer, Part B: Fundamentals, 32(4), pp.369–384.

Natsui, S., Takai, H., Kumagai, T., Kikuchi, T., & Suzuki, R., 2014. Stable mesh-free moving particle semi-implicit method for direct analysis of gas-liquid two-phase flow. Chemical Engineering Science, 111, pp.286–298.

Ng, K.C., Hwang, Y.H., Sheu, T.W.H. & Yu, C.H., 2015. Moving Particle Level-Set (MPLS) method for incompressible multiphase flow computation. Computer Physics Communications.

Noh, W.. & Woodward, P., 1976. SLIC. Energy, pp.1–6.

Odar, F. & Hamilton, W.S., 1964. Forces on a Sphere Accelerating in Pluid. Journal of Fluid Mechanics, 18(2), pp.302–314.

Olsson, E. & Kreiss, G., 2005. A conservative level set method for two phase flow. Journal of Computational Physics, 210(1), pp.225–246.

Osher, S. & Sethian, J.A., 1988. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics, 79, pp.12–49.

Owkes, M. & Desjardins, O., 2013. A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows. Journal of Computational Physics, 249, pp.275–302.

Owkes, M. & Desjardins, O., 2015. A mesh-decoupled height function method for computing interface curvature. Journal of Computational Physics, 281, pp.285–300.

Patankar, N.A. & Joseph, D.D., 2001. Modeling and numerical simulation of particulate flows by the Eulerian-Lagrangian approach. International Journal of Multiphase Flow, 27(10), pp.1659–1684.

Perot, B. & Nallapati, R., 2003. A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows. Journal of Computational Physics, 184(1), pp.192–214.

van der Pijl, S.P., Segal, A., Vuik, C. & Wesseling, P., 2005. A mass-conserving Level-Set method for modelling of multi-phase flows. International Journal for Numerical Methods in Fluids, 47(4), pp.339–361.

Popinet, S., 2009. An accurate adaptive solver for surface-tension-driven interfacial flows.

Journal of Computational Physics, 228(16), pp.5838–5866.

Price, G., Reader, G., Rowe, R. & Bugg, J., 2001. A piecewise parabolic interface calculation for volume-tracking. In 6th Annual conference of the computational fluid dynamics society of Canada. Quebec.

Prosperetti, A., 1981. Motion of two superposed viscous fluids. Physics of fluids, 24(7), pp.1217–1223.

Prosperetti, A. & Tryggvason, G., 2007. Computational Methods for Multiphase Flow, New York: Cambridge University Press.

Quan, S. & Schmidt, D.P., 2007. A moving mesh interface tracking method for 3D incompressible two-phase flows. Journal of Computational Physics, 221(2), pp.761–780.

Raeini, A.Q., Blunt, M.J. & Bijeljic, B., 2012. Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method. Journal of Computational Physics, 231(17), pp.5653–5668.

Raessi, M., Bussmann, M. & Mostaghimi, J., 2009. A semi-implicit finite volume implementation of the CSF method for treating surface tension in interfacial flow. International Journal for Numerical Methods in Fluids, 59, pp.1093–1110.

Rafiee, A., Cummins, S., Rudman, M. & Thiagarajan, K., 2012. Comparative study on the accuracy and stability of SPH schemes in simulating energetic free-surface flows. European Journal of Mechanics, B/Fluids, 36, pp.1–16.

Renardy, M. & Li, J., 2001. Numerical simulation of moving contact line problems using a volume-of-fluid method. Journal of Computational Physics, 171(1), pp.243–263.

Renardy, Y. & Renardy, M., 2002. PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method. Journal of Computational Physics, 183(2), pp.400–421.

Ritchie, B.W. & Thomas, P.A., 2001. Multiphase smoothed-particle hydrodynamics. , 756.

Rottman, J.W. & Simpson, J.E., 1983. Gravity currents produced by instantaneous releases of a heavy fluid in a rectangular channel. Journal of Fluid Mechanics, 135, pp.95–110.

Rudman, M., 1997. Volume-Tracking Methods for Interfacial Flow Calculations. International Journal for Numerical Methods in Fluids, 24(7), pp.671–691.

Sethian, J.A. & Smereka, P., 2003. Level Set Methods for Fluid Interfaces. Annual Review of Fluid Mechanics, 35(1), pp.341–372.

Shakibaeinia, A. & Jin, Y., 2012a. Lagrangian multiphase modeling of sand discharge into still water. Advances in Water Resources, 48, pp.55–67.

Shakibaeinia, A. & Jin, Y., 2012b. MPS mesh-free particle method for multiphase flows. Computer Methods in Applied Mechanics and Engineering, 229-232, pp.13–26.

Shao, S.D., 2012. Incompressible smoothed particle hydrodynamics simulation of

multifluid flows. International Journal for Numerical Methods in Fluids, 69, pp.1715–1735.

Shao, S.D., 2009. Incompressible SPH simulation of water entry of a free-falling object. International Journal for Numerical Methods in Fluids, 59, pp.59–115.

Shao, S.D. & Lo, E.Y.M., 2003. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. Advances in Water Resources, 26(7), pp.787–800.

Sokolichin, A. & Eigenberger, G., 1997. Dynamic numerical simulation of gas-liquid two-phase flows: Euler/Euler versus Euler/Lagrange. Chemical Engineering Science, 52(4), pp.611–626.

Sriram, V. & Ma, Q.W., 2012. Improved MLPG_R method for simulating 2D interaction between violent waves and elastic structures. Journal of Computational Physics, 231(22), pp.7650–7670.

Stone, H.A., 1990. A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface. Physics Of Fluids A Fluid Dynamics, 2(1), pp.111–112.

Sussman, M., Smith, K.M., Hussaini, M.Y., Ohta, M. & Zhi-Wei, R., 2007. A sharp interface method for incompressible two-phase flows. Journal of Computational Physics, 221(2), pp.469–505.

Sussman, M. & Fatemi, E., 1999. An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow. SIAM Journal on Scientific Computing, 20(4), pp.1165–1191.

Swegle, J.W. & Attaway, S.W., 1995. On the feasibility of using smoothed particle hydrodynamics for underwater explosion calculations. Computational Mechanics, 17(3), pp.151–168.

Szewc, K., Pozorski, J. & Minier, J.P., 2015. Spurious interface fragmentation in multiphase SPH K. International Journal for Numerical Methods in Engineering, 103(9), pp.625–649.

Tanaka, M. & Masunaga, T., 2010. Stabilization and smoothing of pressure in MPS method by Quasi-Compressibility. Journal of Computational Physics, 229(11), pp.4279–4290.

Tripathi, M.K., Sahu, K.C. & Govindarajan, R., 2015. Dynamics of an initially spherical bubble rising in quiescent liquid. Nature Communications, 6, p.6268.

Tsuruta, N., Khayyer, a. & Gotoh, H., 2013. A short note on Dynamic Stabilization of Moving Particle Semi-implicit method. Computers and Fluids, 82, pp.158–164.

Tsuruta, N., Khayyer, a. & Gotoh, H., 2015. Space potential particles to enhance the stability of projection-based particle methods. International Journal of Computational Fluid Dynamics, 29(1), pp.100–119.

Tukovic, Z. & Jasak, H., 2012. A moving mesh finite volume interface tracking method

for surface tension dominated interfacial fluid flow. Computers and Fluids, 55, pp.70–84.

Ubbink, O. & Issa, R.I., 1999. A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes. Journal of Computational Physics, 153(1), pp.26–50.

Valizadeh, A. & Monaghan, J.J., 2015. A study of solid wall models for weakly compressible SPH. Journal of Computational Physics, 300, pp.5–19.

Velazquez, P.F., Gomez, D.O., Dubner, G.M., de Castro, G.G. & Costa, A., 1998. Study of the Rayleigh-Taylor instability in Tycho's supernova remnant. Astronomy and Astrophysics, 334, pp.1060–1067.

Walke, S.M. & Sathe, V.S., 2012. Experimental Study on Comparison of Rising Velocity of Bubbles and Light Weight Particles in the Bubble Column. International Journal of Chemical Engineering and Applications, 3(1), pp.25–30.

Wang, Z. & Tong, A.Y., 2010. A sharp surface tension modeling method for two-phase incompressible interfacial flows. International Journal for Numerical Methods in Fluids, (64), pp.709–732.

Weller, H.G., 2008. A New Approach to VOF-based Interface Capturing Methods for Incompressible and Compressible Flow. Technical Report TR/HGW/07, OpenCFD Ltd., (May).

Wörner, M., 2012. Numerical modeling of multiphase flows in microfluidics and micro process engineering: A review of methods and applications. Microfluidics and Nanofluidics, 12(6), pp.841–886.

Xiao, F., Honma, Y. & Kono, T., 2005. A simple algebraic interface capturing scheme using hyperbolic tangent function. International Journal for Numerical Methods in Fluids, 48(9), pp.1023–1040.

Xu, J.J., Li, Z., Lowengrub, J. & Zhao, H., 2006. A level-set method for interfacial flows with surfactant. Journal of Computational Physics, 212(2), pp.590–616.

Xu, T. & Jin, Y., 2014. Numerical investigation of flow in pool-and-weir fishways using a meshless particle method. Journal of Hydraulic Research, 52(6), pp.849–861.

Xue, Q., Dalluge, D., Heindel, T.J., Fox, R.O. & Brown, R.C., 2012. Experimental validation and CFD modeling study of biomass fast pyrolysis in fluidized-bed reactors. Fuel, 97, pp.757–769.

Yokoi, K., 2007. Efficient implementation of THINC scheme: A simple and practical smoothed VOF algorithm. Journal of Computational Physics, 226(2), pp.1985–2002.

Yoon, H.Y., Koshizuka, S. & Oka, Y., 2001. Direct calculation of bubble growth, departure, and rise in nucleate pool boiling. International Journal of Multiphase Flow, 27(2), pp.277–298.

Youngs, D.L., 1982. Time-dependent multi-material flow with large fluid distortion. In M. K.W. & B. M.J., eds. Numerical methods for fluid dynamics. New York: Academic Press, pp. 273–285.

Zainali, A., Tofighi, N., Shadloo, M.S. & Yildiz, M., 2013. Numerical investigation of Newtonian and non-Newtonian multiphase flows using ISPH method. Computer Methods in Applied Mechanics and Engineering, 254, pp.99–113.

Zalesak, S., 1979. Fully multidimentional flux-corrected transport algorithms for fluids. Journal of Computational Physics, 31(3), pp.335–362.

Zhang, M., 2010. Simulation of surface tension in 2D and 3D with smoothed particle hydrodynamics method. Journal of Computational Physics, 229(19), pp.7238–7259.

Zhang, M., Zhang, S., Zhang, H. & Zheng, L., 2012. Simulation of surface-tension-driven interfacial flow with smoothed particle hydrodynamics method. Computers and Fluids, 59, pp.61–71.

Zhang, Y., Zou, Q. & Greaves, D., 2010. Numerical simulation of free-surface flowusing the level-set method with global mass correction Yali. International Journal for Numerical Methods in Fluids, 63, pp.651–680.

Zheng, X., Ma, Q.W. & Duan, W.Y., 2014. Incompressible SPH method based on Rankine source solution for violent water wave simulation. Journal of Computational Physics, 276, pp.291–314.

Zhou, C., Yue, P., Feng, J.J., Ollivier-Gooch, C.F. & Hu, H.H., 2010. 3D phase-field simulations of interfacial dynamics in Newtonian and viscoelastic fluids. Journal of Computational Physics, 229(2), pp.498–511.

Zhou, J.T. & Ma, Q.W., 2010. MLPG Method Based on Rankine Source Solution for Modelling 3D BreakingWaves. Computer Modeling in Engineering and, 56(2), pp.179–210.